



La guía de AWS CDK capas

# AWS Guía prescriptiva



# AWS Guía prescriptiva: La guía de AWS CDK capas

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Las marcas comerciales y la imagen comercial de Amazon no se pueden utilizar en relación con ningún producto o servicio que no sea de Amazon, de ninguna manera que pueda causar confusión entre los clientes y que menosprecie o desacredite a Amazon. Todas las demás marcas registradas que no son propiedad de Amazon son propiedad de sus respectivos propietarios, que pueden o no estar afiliados, conectados o patrocinados por Amazon.

---

# Table of Contents

Introducción .....	1
Construcciones de capa 1 .....	3
El AWS CDK... CloudFormation ciclo de vida de las construcciones de L1 .....	3
La especificación del AWS CloudFormation recurso .....	4
Construcciones de capa 2 .....	7
Propiedades predeterminadas .....	9
Estructuras, tipos e interfaces .....	10
Métodos estáticos .....	10
Métodos auxiliares .....	11
Enums .....	13
Clases auxiliares .....	13
Construcciones de capa 3 .....	15
Interacciones de recursos .....	16
Extensiones de recursos .....	18
Recursos personalizados .....	19
Prácticas recomendadas .....	28
Preguntas frecuentes .....	30
¿No puedo usar las capas AWS CDK sin entender las capas? .....	30
¿Puedo crear construcciones de L2 a partir de L1 de la misma manera que hago construcciones de L3 a partir de L2? .....	30
¿Qué AWS recursos aún no tienen construcciones oficiales de L2? .....	30
¿Puedo hacer una construcción de L2 o L3 en cualquier idioma que sea compatible? AWS CDK .....	30
¿Dónde puedo encontrar las construcciones de L3 existentes fuera de? AWS CDK .....	31
Recursos .....	32
Historial de documentos .....	33
Glosario .....	34
# .....	34
A .....	35
B .....	38
C .....	40
D .....	43
E .....	47
F .....	50

---

G .....	52
H .....	53
I .....	54
L .....	57
M .....	58
O .....	62
P .....	65
Q .....	68
R .....	68
S .....	71
T .....	75
U .....	77
V .....	78
W .....	78
Z .....	79
.....	lxxxi

# La guía AWS CDK de capas

Steven Guggenheimer, Amazon Web Services (AWS)

Diciembre de 2023 ([historial de documentos](#))

Uno de los conceptos principales en los que AWS Cloud Development Kit (AWS CDK) se basa es muy parecido al concepto de mantenerse caliente en un día frío. Ese concepto se llama estratificación. En un día frío te pones una camisa, una chaqueta y, a veces, una chaqueta aún más grande dependiendo del frío que haga. Luego, si entras y la calefacción está encendida, puedes quitarte una o ambas capas de la chaqueta para que no haga demasiado calor. AWS CDK Utiliza capas para proporcionar diferentes niveles de abstracción al usar componentes de la nube. La estratificación garantiza que nunca tenga que escribir demasiado código ni tener muy poco acceso a las propiedades de los recursos al implementar su infraestructura como pilas de código (IAC).

Si no las utilizas AWS CDK, tendrás que escribir las [AWS CloudFormation](#) plantillas a mano; es decir, utilizarás solo una capa, lo que te obligará a escribir mucho más código del que normalmente es necesario. Por otro lado, si AWS CDK tuviera que abstraer todo lo CloudFormation que normalmente no necesitas escribir, no podrías gestionar ningún caso extremo.

Para solucionar este problema, AWS CDK divide el aprovisionamiento de recursos en tres capas separadas y distintas:

- Capa 1: la CloudFormation capa: la capa más básica en la que el CloudFormation recurso y el AWS CDK recurso son casi idénticos.
- Capa 2: la capa seleccionada: la capa en la que CloudFormation los recursos se abstraen en clases programáticas que simplifican gran parte de la sintaxis repetitiva que se utiliza CloudFormation en su interior. Esta capa constituye la mayor parte de. AWS CDK
- Capa 3: la capa de patrón: la capa más abstracta en la que puede usar los componentes básicos proporcionados por las capas 1 y 2 para personalizar el código para su caso de uso específico.

Cada elemento de cada capa es una instancia de una AWS CDK clase especial llamada `Construct`. Según la [AWS documentación](#), las construcciones son «los componentes básicos de AWS CDK las aplicaciones. Una construcción representa un «componente de nube» y encapsula todo lo AWS CloudFormation necesario para crear el componente». Las construcciones dentro de estas capas se conocen como construcciones L1, L2 y L3 según la capa a la que pertenezcan. En

esta guía, haremos un recorrido por cada AWS CDK capa para averiguar para qué se utilizan y por qué son importantes.

Esta guía está dirigida a los gerentes técnicos, líderes y desarrolladores que estén interesados en profundizar en los conceptos básicos que hacen que AWS CDK funcione. AWS CDK Es una herramienta popular, pero es muy común que los equipos se pierdan gran parte de lo que ofrece. Cuando comiences a entender los conceptos descritos en esta guía, podrás descubrir un mundo completamente nuevo de posibilidades y optimizar los procesos de aprovisionamiento de recursos de tus equipos.

En esta guía:

- [Construcciones de capa 1](#)
- [Construcciones de capa 2](#)
- [Construcciones de capa 3](#)
- [Prácticas recomendadas](#)
- [PREGUNTAS FRECUENTES](#)
- [Recursos](#)

# Construcciones de capa 1

Los [constructos de L1 son los](#) componentes básicos del AWS CDK y se distinguen fácilmente de otros constructos por el prefijo. Cfn Por ejemplo, el paquete Amazon DynamoDB del contiene `Table` una construcción, que es una construcción L2. AWS CDK Se llama `CfnTable` a la construcción L1 correspondiente y representa directamente una CloudFormation `Table` DynamoDB. Es imposible utilizarla AWS CDK sin acceder a esta primera capa, aunque una AWS CDK aplicación normalmente nunca utiliza una construcción L1 directamente. Sin embargo, en la mayoría de los casos, las construcciones L2 y L3 que los desarrolladores están acostumbrados a usar dependen en gran medida de las construcciones L1. Por lo tanto, puede pensar en las construcciones de L1 como el puente entre y. CloudFormation AWS CDK

El único propósito de AWS CDK es generar CloudFormation plantillas mediante el uso de lenguajes de codificación estándar. Tras ejecutar el comando CLI `cdk synth` y generar las CloudFormation plantillas resultantes, el trabajo AWS CDK estará completo. El comando `cdk deploy` está ahí solo por comodidad, pero lo que se hace cuando se ejecuta ese comando es totalmente interno. CloudFormation La pieza del rompecabezas que traduce el AWS CDK código a un formato que CloudFormation comprenda es la construcción L1.

## El AWS CDK... CloudFormation ciclo de vida de las construcciones de L1

El proceso de creación y uso de las construcciones de L1 consta de los siguientes pasos:

1. El proceso de AWS CDK construcción convierte CloudFormation las especificaciones en código programático en forma de construcciones de L1.
2. Los desarrolladores escriben código que hace referencia directa o indirectamente a las construcciones de L1 como parte de una aplicación. AWS CDK
3. Los desarrolladores ejecutan el comando `cdk synth` para volver a convertir el código programático al formato dictado por las especificaciones (plantillas). CloudFormation
4. Los desarrolladores ejecutan el comando `cdk deploy` para implementar las CloudFormation pilas de estas plantillas en los entornos de cuentas. AWS

Hagamos un pequeño ejercicio. Ve al [repositorio de código AWS CDK abierto](#) de GitHub, elige un AWS servicio aleatorio y, a continuación, ve al AWS CDK paquete correspondiente a ese servicio

(ubicado en la carpeta `packages,aws-cdk-lib,aws-<servicename>,lib`). Para este ejemplo, escogamos Amazon S3, pero funciona para cualquier servicio. Si observa el [archivo `index.ts`](#) principal de ese paquete, verá una línea que dice:

```
export * from './s3.generated';
```

Sin embargo, no verá el `s3.generated` archivo en ningún lugar del directorio correspondiente. Esto se debe a que las construcciones L1 se generan automáticamente a partir de la [especificación del CloudFormation recurso](#) durante el AWS CDK proceso de compilación. Esto solo lo verá `s3.generated` en el paquete después de ejecutar el comando de AWS CDK compilación del paquete.

## La especificación del AWS CloudFormation recurso

La especificación del AWS CloudFormation recurso define la infraestructura como código (IAC) AWS y determina cómo el código de CloudFormation las plantillas se convierte en recursos en una AWS cuenta. Esta especificación define AWS los recursos en [formato JSON](#) por región. A cada recurso se le asigna un [nombre de tipo de recurso](#) único que sigue el formato `provider::service::resource`. Por ejemplo, el nombre del tipo de recurso de un bucket de Amazon S3 sería `AWS::S3::Bucket` y el nombre del tipo de recurso de un punto de acceso de Amazon S3 sería `AWS::S3::AccessPoint`. Estos tipos de recursos se pueden representar en una CloudFormation plantilla mediante la sintaxis definida en la especificación del AWS CloudFormation recurso. Cuando se ejecuta el proceso de AWS CDK compilación, cada tipo de recurso también se convierte en una construcción L1.

En consecuencia, cada construcción de L1 es una imagen especular programática de su recurso correspondiente. CloudFormation Todas las propiedades que se aplicarían en una CloudFormation plantilla están disponibles al crear una instancia de una construcción de L1, y todas las CloudFormation propiedades necesarias también se requieren como argumento al crear una instancia de la construcción de L1 correspondiente. En la siguiente tabla, se compara un depósito de S3 representado en una CloudFormation plantilla con el mismo depósito de S3 definido como construcción de L1. AWS CDK

### CloudFormation plantilla

```
"amzns3demobucket": {  
  "Type": "AWS::S3::Bucket",
```

### constructo L1

```
new CfnBucket(this, "amzns3de  
mobucket", {
```

```

    "Properties": {
      "BucketName": "amzn-s3-demo-
bucket",
      "BucketEncryption": {
        "ServerSideEncryptionConfig
uration": [
          {
            "ServerSideEncrypt
ionByDefault": {
              "SSEAlgorithm": "AES256"
            }
          }
        ],
        "MetricsConfigurations": [
          {
            "Id": "myConfig"
          }
        ],
        "OwnershipControls": {
          "Rules": [
            {
              "ObjectOwnership":
"BucketOwnerPreferred"
            }
          ]
        },
        "PublicAccessBlockConfigura
tion": {
          "BlockPublicAcls": true,
          "BlockPublicPolicy": true,
          "IgnorePublicAcls": true,
          "RestrictPublicBuckets": true
        },
        "VersioningConfiguration": {
          "Status": "Enabled"
        }
      }
    }
  }
}

```

```

    bucketName: "amzn-s3-demo-bucket",
    bucketEncryption: {
      serverSideEncryptionConfigu
ration: [
        {
          serverSideEncryptionByDefau
lt: {
            sseAlgorithm: "AES256"
          }
        }
      ],
      metricsConfigurations: [
        {
          id: "myConfig"
        }
      ],
      ownershipControls: {
        rules: [
          {
            objectOwnership: "BucketOw
nerPreferred"
          }
        ]
      },
      publicAccessBlockConfiguration: {
        blockPublicAcls: true,
        blockPublicPolicy: true,
        ignorePublicAcls: true,
        restrictPublicBuckets: true
      },
      versioningConfiguration: {
        status: "Enabled"
      }
    });

```

Como puede ver, la construcción L1 es la manifestación exacta en el código del CloudFormation recurso. No hay atajos ni simplificaciones, por lo que la cantidad de texto repetitivo que debe

escribirse es aproximadamente la misma. Sin embargo, se supone que una de las grandes ventajas de usar el AWS CDK es que ayuda a eliminar gran parte de esa sintaxis repetitiva. CloudFormation Entonces, ¿cómo sucede eso? Ahí es donde entra en juego la construcción L2.

## Construcciones de capa 2

El [repositorio de código AWS CDK abierto](#) se escribe principalmente con el lenguaje de [TypeScript](#) programación y consta de numerosos paquetes y módulos. La biblioteca de paquetes principal `aws-cdk-lib`, denominada `aws-cdk-lib`, se divide aproximadamente en un paquete por AWS servicio, aunque no siempre es así. Como se mencionó anteriormente, las construcciones L1 se generan automáticamente durante el proceso de compilación, entonces, ¿qué es todo ese código que ves cuando miras dentro del repositorio? Se trata de [construcciones L2, que son abstracciones de las construcciones de L1](#).

Los paquetes también contienen una colección de TypeScript tipos, enumeraciones e interfaces, así como clases auxiliares que añaden más funcionalidad, pero todos esos elementos sirven para construcciones de L2. Todas las construcciones de L2 llaman a sus correspondientes construcciones de L1 en sus constructores tras la instanciación, y se puede acceder a la construcción de L1 resultante que se crea desde la capa 2 de la siguiente manera:

```
const role = new Bucket(this, "amzn-s3-demo-bucket", {/*...BucketProps*/});
const cfnBucket = role.node.defaultChild;
```

La construcción L2 toma las propiedades predeterminadas, los métodos prácticos y otros elementos sintácticos y los aplica a la construcción L1. Esto elimina gran parte de la repetición y la verbosidad necesarias para aprovisionar recursos directamente. CloudFormation

Todos los constructos L2 construyen sus correspondientes constructos L1 bajo el capó. Sin embargo, las construcciones L2 en realidad no amplían las construcciones L1. [Tanto las construcciones L1 como las L2 heredan una clase especial denominada Construct](#). [En la versión 1, AWS CDK la Construct clase estaba integrada en el kit de desarrollo, pero en la versión 2 es un paquete independiente independiente](#). Esto es para que otros paquetes, como el [Cloud Development Kit for Terraform \(CDKTF\)](#), puedan incluirlo como una dependencia. Cualquier clase que herede la Construct clase es una construcción L1, L2 o L3. Las construcciones L2 amplían esta clase directamente, mientras que las construcciones L1 extienden una clase denominada `CfnResource`, como se muestra en la siguiente tabla.

Árbol de herencia de L1

Construcción L1

→ clase [CfnResource](#)

Árbol de herencia de L2

Construcción L2

→ [clase Construct](#)

→ clase abstracta [CfnRefElement](#)

→→→ clase abstracta [CfnElement](#)

→→→→ [clase Construct](#)

Si tanto las construcciones L1 como las L2 heredan la Construct clase, ¿por qué las construcciones L2 no amplían simplemente L1? Bueno, las clases entre la Construct clase y la capa 1 bloquean la construcción L1 como una imagen reflejada del recurso. CloudFormation Contienen métodos abstractos (métodos que deben incluir las clases posteriores)\_toCloudFormation, que obligan a la construcción a generar directamente la sintaxis de salida CloudFormation. Las construcciones de L2 omiten esas clases y las Construct amplían directamente. Esto les da la flexibilidad de abstraer gran parte del código necesario para las construcciones de L1 al crearlas por separado dentro de sus constructores.

En la sección anterior, se side-by-side comparó un depósito de S3 a partir de una CloudFormation plantilla y ese mismo depósito de S3 renderizado como una construcción de L1. Esa comparación mostró que las propiedades y la sintaxis son casi idénticas, y que la construcción L1 guarda solo tres o cuatro líneas en comparación con la CloudFormation construcción. Ahora comparemos la construcción L1 con la construcción L2 para el mismo segmento de S3:

### Construcción L1 para el cubo S3

```
new CfnBucket(this, "amzn3demobucket", {
  bucketName: "amzn-s3-demo-bucket",
  bucketEncryption: {
    serverSideEncryptionConfiguration: [
      {
        serverSideEncryptionByDefault: {
          sseAlgorithm: "AES256"
        }
      }
    ]
  },
  metricsConfigurations: [
    {
```

### Construcción L2 para cubo S3

```
new Bucket(this, "amzn3demobucket",
{
  bucketName: "amzn-s3-demo-bucket",
  encryption: BucketEncryption.S3_MANAGED,
  metrics: [
    {
      id: "myConfig"
    }
  ],
  objectOwnership: ObjectOwnership.BUCKET_OWNER_PREFERRED,
  blockPublicAccess: BlockPublicAccess.BLOCK_ALL,
  versioned: true
});
```

```
        id: "myConfig"
    }
],
ownershipControls: {
    rules: [
        {
            objectOwnership: "BucketOwnerPreferred"
        }
    ]
},
publicAccessBlockConfiguration: {
    blockPublicAcls: true,
    blockPublicPolicy: true,
    ignorePublicAcls: true,
    restrictPublicBuckets: true
},
versioningConfiguration: {
    status: "Enabled"
}
});
```

Como puede ver, la construcción L2 tiene menos de la mitad del tamaño de la construcción L1. Las construcciones L2 utilizan numerosas técnicas para lograr esta consolidación. Algunas de estas técnicas se aplican a un solo constructo L2, pero otras se pueden reutilizar en varios constructos para separarlos en su propia clase para su reutilización. Las construcciones de L2 consolidan la CloudFormation sintaxis de varias maneras, como se explica en las siguientes secciones.

## Propiedades predeterminadas

La forma más sencilla de consolidar el código para aprovisionar un recurso es convertir la configuración de propiedades más común en valores predeterminados. AWS CDK Tiene acceso a potentes lenguajes de programación y CloudFormation no los tiene, por lo que estos valores predeterminados suelen ser de naturaleza condicional. A veces, se pueden eliminar varias líneas de CloudFormation configuración del AWS CDK código porque esas configuraciones se pueden deducir de los valores de otras propiedades que se pasan a la construcción.

## Estructuras, tipos e interfaces

Aunque AWS CDK está disponible en varios lenguajes de programación, está escrito de forma nativa TypeScript, por lo que el sistema de tipos de ese lenguaje se utiliza para definir los tipos que componen las construcciones L2. Profundizar en ese sistema de tipos va más allá del alcance de esta guía; consulte la [TypeScript documentación](#) para obtener más información. En resumen, a TypeScript `type` describe qué tipo de datos contiene una variable en particular. Pueden ser datos básicos, como `unstring`, o datos más complejos, como `unobject`. A TypeScript `interface` es otra forma de expresar el tipo de TypeScript objeto y a `struct` es otro nombre para una interfaz.

TypeScript no usa el término estructura, pero si consulta la [referencia de la AWS CDK API](#), verá que una estructura es en realidad solo otra TypeScript interfaz dentro del código. La referencia de la API también se refiere a determinadas interfaces como interfaces. Si las estructuras y las interfaces son lo mismo, ¿por qué la AWS CDK documentación hace una distinción entre ellas?

Lo que AWS CDK se denomina estructuras son interfaces que representan cualquier objeto utilizado por una construcción L2. Esto incluye los tipos de objetos de los argumentos de propiedad que se pasan a la construcción L2 durante la instanciación, como `BucketProps` la construcción S3 Bucket y `TableProps` la construcción Tabla de DynamoDB, así como otras TypeScript interfaces que se utilizan en. AWS CDK En resumen, si se trata de una TypeScript interfaz dentro de AWS CDK y su nombre no lleva el prefijo de la letra *I*, se denomina estructura. AWS CDK

Por el contrario, AWS CDK utiliza el término interfaz para representar los elementos básicos; un objeto simple debería considerarse una representación adecuada de una construcción o clase auxiliar en particular. Es decir, una interfaz describe cuáles deben ser las propiedades públicas de una construcción L2. Todos los nombres de las AWS CDK interfaces son los nombres de las construcciones o clases auxiliares existentes con el prefijo de la letra *I*. Todas las construcciones de L2 amplían la `Construct` clase, pero también implementan su interfaz correspondiente. Por lo tanto, la construcción L2 Bucket implementa la interfaz `IBucket`

## Métodos estáticos

Cada instancia de una construcción L2 también es una instancia de su interfaz correspondiente, pero no ocurre lo contrario. Esto es importante a la hora de analizar una estructura para ver qué tipos de datos son necesarios. Si una estructura tiene una propiedad llamada `bucket`, que requiere el tipo de datos `IBucket`, puede pasar un objeto que contenga las propiedades enumeradas en la `IBucket` interfaz o una instancia de una capa 2. `Bucket` Cualquiera de las dos funcionaría. Sin embargo, si

esa bucket propiedad requiriera una `L2Bucket`, solo podrías pasar una `Bucket` instancia en ese campo.

Esta distinción se vuelve muy importante cuando importas recursos preexistentes a tu pila. Puedes crear una construcción de L2 para cualquier recurso que sea nativo de tu pila, pero si necesitas hacer referencia a un recurso que se creó fuera de la pila, tienes que usar la interfaz de esa construcción de L2. Esto se debe a que al crear una construcción de L2 se crea un recurso nuevo si aún no existe uno dentro de esa pila. Las referencias a los recursos existentes deben ser objetos simples que se ajusten a la interfaz de esa construcción de L2.

Para facilitar esto en la práctica, la mayoría de las construcciones de L2 tienen un conjunto de métodos estáticos asociados que devuelven la interfaz de esa construcción de L2. Estos métodos estáticos suelen empezar con la palabra `from`. Los dos primeros argumentos que se pasan a estos métodos son los mismos `scope` y `id` son necesarios para una construcción L2 estándar. Sin embargo, el tercer argumento no `props` es más que un pequeño subconjunto de propiedades (o, a veces, solo una propiedad) que define una interfaz. Por esta razón, cuando se pasa una construcción L2, en la mayoría de los casos solo se requieren los elementos de la interfaz. Esto permite utilizar también los recursos importados, siempre que sea posible.

```
// Example of referencing an external S3 bucket
const preExistingBucket = Bucket.fromBucketName(this, "external-bucket", "name-of-
bucket-that-already-exists");
```

Sin embargo, no debes confiar demasiado en las interfaces. Debe importar los recursos y usar las interfaces directamente solo cuando sea absolutamente necesario, ya que las interfaces no proporcionan muchas de las propiedades (como los métodos de ayuda) que hacen que una construcción de L2 sea tan potente.

## Métodos auxiliares

Una construcción L2 es una clase programática y no un objeto simple, por lo que puede exponer métodos de clase que permiten manipular la configuración de los recursos una vez realizada la instanciación. [Un buen ejemplo de ello es la construcción de roles L2 AWS Identity and Access Management \(IAM\)](#). Los siguientes fragmentos muestran dos formas de crear el mismo rol de IAM mediante la construcción L2. `Role`

Sin un método auxiliar:

```
const role = new Role(this, "my-iam-role", {
```

```

    assumedBy: new FederatedPrincipal('my-identity-provider.com'),
    managedPolicies: [
      ManagedPolicy.fromAwsManagedPolicyName("ReadOnlyAccess")
    ],
    inlinePolicies: {
      lambdaPolicy: new PolicyDocument({
        statements: [
          new PolicyStatement({
            effect: Effect.ALLOW,
            actions: [ 'lambda:UpdateFunctionCode' ],
            resources: [ 'arn:aws:lambda:us-east-1:123456789012:function:my-
function' ]
          })
        ]
      })
    }
  });

```

Con un método auxiliar:

```

const role = new Role(this, "my-iam-role", {
  assumedBy: new FederatedPrincipal('my-identity-provider.com')
});

role.addManagedPolicy(ManagedPolicy.fromAwsManagedPolicyName("ReadOnlyAccess"));
role.attachInlinePolicy(new Policy(this, "lambda-policy", {
  policyName: "lambdaPolicy",
  statements: [
    new PolicyStatement({
      effect: Effect.ALLOW,
      actions: [ 'lambda:UpdateFunctionCode' ],
      resources: [ 'arn:aws:lambda:us-east-1:123456789012:function:my-function' ]
    })
  ]
}));

```

La capacidad de utilizar métodos de instancia para manipular la configuración de los recursos después de la instanciación proporciona a las construcciones de L2 mucha flexibilidad adicional con respecto a la capa anterior. Las construcciones de nivel 1 también heredan algunos métodos de recursos (por ejemplo `addPropertyOverride`), pero no se obtienen métodos diseñados específicamente para ese recurso y sus propiedades hasta la segunda capa.

## Enums

CloudFormation La sintaxis suele requerir que se especifiquen muchos detalles para poder aprovisionar un recurso correctamente. Sin embargo, la mayoría de los casos de uso suelen estar cubiertos por solo un puñado de configuraciones. La representación de esas configuraciones mediante una serie de valores enumerados puede reducir considerablemente la cantidad de código necesaria.

Por ejemplo, en el ejemplo de código L2 del bucket S3 descrito anteriormente en esta sección, debe usar la `bucketEncryption` propiedad de la CloudFormation plantilla para proporcionar todos los detalles, incluido el nombre del algoritmo de cifrado que se va a utilizar. En su lugar, AWS CDK proporciona la `BucketEncryption` enumeración, que utiliza las cinco formas más comunes de cifrado por cubos y permite expresar cada una de ellas mediante nombres de variables individuales.

¿Qué pasa con los casos extremos que no están incluidos en las enumeraciones? Uno de los objetivos de una construcción de nivel 2 es simplificar la tarea de aprovisionamiento de un recurso de capa 1, por lo que es posible que algunos casos extremos que se utilizan con menos frecuencia no se admitan en la capa 2. Para respaldar estos casos extremos, AWS CDK permite manipular las propiedades de los CloudFormation recursos subyacentes directamente mediante el [addPropertyOverride](#) método. Para obtener más información sobre las anulaciones de propiedades, consulte la sección de [prácticas recomendadas](#) de esta guía y la sección [Abstracciones y trampillas de escape](#) de la documentación. AWS CDK

## Clases auxiliares

A veces, una enumeración no puede cumplir con la lógica programática necesaria para configurar un recurso para un caso de uso determinado. En estas situaciones, AWS CDK a menudo se ofrece una clase auxiliar en su lugar. Una enumeración es un objeto simple que ofrece una serie de pares clave-valor, mientras que una clase auxiliar ofrece todas las capacidades de una clase. TypeScript Una clase auxiliar aún puede actuar como una enumeración al exponer propiedades estáticas, pero esas propiedades podrían tener sus valores establecidos internamente con lógica condicional en el constructor de la clase auxiliar o en un método auxiliar.

Por lo tanto, aunque la `BucketEncryption` enumeración puede reducir la cantidad de código necesaria para configurar un algoritmo de cifrado en un bucket de S3, esa misma estrategia no funcionaría para establecer la duración del tiempo, ya que hay demasiados valores posibles entre los que elegir. Crear una enumeración para cada valor sería mucho más difícil de lo que vale la pena. Por este motivo, se utiliza una clase auxiliar para los ajustes de configuración predeterminados

de S3 Object Lock de un bucket de S3, tal como los representa la [ObjectLockRetention](#) clase. `ObjectLockRetention` contiene dos métodos estáticos: uno para la retención de la conformidad y otro para la retención de la gobernanza. Ambos métodos utilizan una instancia de la [clase auxiliar Duration](#) como argumento para expresar el tiempo durante el que se debe configurar el bloqueo.

[Otro ejemplo es la clase AWS Lambda auxiliar Runtime](#). A primera vista, puede parecer que las propiedades estáticas asociadas a esta clase podrían gestionarse mediante una enumeración. Sin embargo, a simple vista, el valor de cada propiedad representa una instancia de la propia `Runtime` clase, por lo que la lógica utilizada en el constructor de la clase no podría lograrse dentro de una enumeración.

## Construcciones de capa 3

Si las construcciones L1 traducen literalmente los CloudFormation recursos a código programático y las construcciones L2 sustituyen gran parte de la CloudFormation sintaxis detallada por métodos auxiliares y lógica personalizada, ¿qué hacen las construcciones L3? La respuesta está limitada únicamente por su imaginación. Puede crear la capa 3 para que se adapte a cualquier caso de uso específico. Si su proyecto necesita un recurso que tenga un subconjunto específico de propiedades, puede crear una construcción L3 reutilizable para satisfacer esa necesidad.

Las construcciones L3 se denominan patrones dentro de. AWS CDK Un patrón es cualquier objeto que extiende la Construct clase AWS CDK (o extiende una clase que extiende la Construct clase) para realizar cualquier lógica abstracta más allá de la capa 2. Cuando usa la AWS CDK CLI para ejecutar `cdk init` e iniciar un nuevo AWS CDK proyecto, debe elegir entre tres tipos de AWS CDK aplicaciones: `app`, `lib`, y `sample-app`

```
Available templates:
* app: Template for a CDK Application
  └─ cdk init app --language=[csharp|fsharp|go|java|javascript|python|typescript]
* lib: Template for a CDK Construct Library
  └─ cdk init lib --language=typescript
* sample-app: Example CDK Application with some constructs
  └─ cdk init sample-app --language=[csharp|fsharp|go|java|javascript|python|typescript]
```

`app` y `sample-app` ambas representan AWS CDK aplicaciones clásicas en las que se crean e implementan CloudFormation pilas en entornos de AWS. Si eliges `lib`, eliges construir una estructura L3 completamente nueva. `app` y `sample-app` permiten elegir cualquier idioma que admitan, pero solo puedes elegir TypeScript con `lib` él. AWS CDK Esto se debe a que AWS CDK está escrito de forma nativa TypeScript y utiliza un sistema de código abierto llamado [JSII](#) para traducir el código original a los demás idiomas compatibles. Cuando decides `lib` iniciar tu proyecto, eliges crear una extensión para el AWS CDK.

Cualquier clase que amplíe la Construct clase puede ser una construcción de capa 3, pero los casos de uso más comunes de la capa 3 son las interacciones de recursos, las extensiones de recursos y los recursos personalizados. La mayoría de las construcciones de L3 utilizan uno o más de estos tres casos para ampliar la funcionalidad. AWS CDK

## Interacciones de recursos

Por lo general, una solución emplea varios servicios de AWS que funcionan juntos. Por ejemplo, una CloudFront distribución de Amazon suele utilizar un bucket de S3 como origen y AWS WAF como protección contra las vulnerabilidades más comunes. AWS AppSync y Amazon API Gateway suelen utilizar tablas de Amazon DynamoDB como fuentes de datos para sus APIs. Una canalización AWS CodePipeline suele utilizar Amazon S3 como fuente y AWS CodeBuild para sus etapas de creación. En estos casos, suele ser útil crear una única construcción de L3 que se encargue del aprovisionamiento de dos o más construcciones de L2 interconectadas.

A continuación, se muestra un ejemplo de una construcción L3 que aprovisiona una CloudFront distribución junto con su origen S3, un AWS WAF registro de Amazon Route 53 y un certificado AWS Certificate Manager (ACM) para añadir un punto final personalizado con cifrado en tránsito, todo en una construcción reutilizable:

```
// Define the properties passed to the L3 construct
export interface CloudFrontWebsiteProps {
  distributionProps: DistributionProps
  bucketProps: BucketProps
  wafProps: CfnWebAclProps
  zone: IHostedZone
}

// Define the L3 construct
export class CloudFrontWebsite extends Construct {
  public distribution: Distribution

  constructor(
    scope: Construct,
    id: string,
    props: CloudFrontWebsiteProps
  ) {
    super(scope, id);

    const certificate = new Certificate(this, "Certificate", {
      domainName: props.zone.zoneName,
      validation: CertificateValidation.fromDns(props.zone)
    });

    const defaultBehavior = {
      origin: new S3Origin(new Bucket(this, "bucket", props.bucketProps))
    }
  }
}
```

```
const waf = new CfnWebACL(this, "waf", props.wafProps);
this.distribution = new Distribution(this, id, {
  ...props.distributionProps,
  defaultBehavior,
  certificate,
  domainNames: [this.domainName],
  webAclId: waf.attrArn,
});
}
}
```

Tenga en cuenta que CloudFront Amazon S3, Route 53 y ACM utilizan construcciones L2, pero la ACL web (que define las reglas para gestionar las solicitudes web) usa una construcción L1. Esto se debe a que AWS CDK se trata de un paquete de código abierto en evolución que no está completamente completo y aún no existe una construcción L2. WebAc1 Sin embargo, cualquiera puede contribuir a ello AWS CDK creando nuevas construcciones de L2. Por lo tanto, hasta que no AWS CDK ofrezca una construcción L2WebAc1, tendrá que usar una construcción L1. Para crear un nuevo sitio web mediante la construcción L3CloudFrontWebsite, utilice el siguiente código:

```
const siteADotCom = new CloudFrontWebsite(stack, "siteA", siteAProps);
const siteBDotCom = new CloudFrontWebsite(stack, "siteB", siteBProps);
const siteCDotCom = new CloudFrontWebsite(stack, "siteC", siteCProps);
```

En este ejemplo, la construcción CloudFront Distribution L2 se expone como una propiedad pública de la construcción L3. Seguirán existiendo casos en los que sea necesario exponer propiedades de L3 como esta, según sea necesario. De hecho, lo Distribution volveremos a ver más adelante, en la sección de [recursos personalizados](#).

AWS CDK Incluye algunos ejemplos de patrones de interacción de recursos como este. Además aws-ecs del paquete que contiene las construcciones L2 para Amazon Elastic Container Service (Amazon ECS), tiene un paquete llamado. AWS CDK [aws-ecs-patterns](#) Este paquete contiene varias construcciones de nivel 3 que combinan Amazon ECS con balanceadores de carga de aplicaciones, balanceadores de carga de red y grupos objetivo, al tiempo que ofrece diferentes versiones predefinidas para Amazon Elastic Compute Cloud (Amazon) y. EC2 AWS Fargate Dado que muchas aplicaciones sin servidor utilizan Amazon ECS únicamente con Fargate, estas construcciones de nivel 3 ofrecen una comodidad que permite a los desarrolladores ahorrar tiempo y dinero a los clientes.

## Extensiones de recursos

Algunos casos de uso requieren que los recursos tengan una configuración predeterminada específica que no sea nativa de la construcción L2. A nivel de pila, esto se puede gestionar mediante [aspectos](#), pero otra forma práctica de dotar a una construcción de L2 de nuevos valores predeterminados consiste en ampliar la capa 2. Como un constructo es cualquier clase que hereda la `Construct` clase y los constructos L2 amplían esa clase, también se puede crear un constructo L3 extendiendo directamente un constructo L2.

Esto puede resultar especialmente útil para la lógica empresarial personalizada que respalde las necesidades específicas de un cliente. Supongamos que una empresa tiene un repositorio que almacena todo su código de AWS Lambda función en un único directorio llamado `src/lambda` y que la mayoría de las funciones de Lambda reutilizan el mismo tiempo de ejecución y el mismo nombre de controlador cada vez. En lugar de configurar la ruta del código cada vez que configure una nueva función Lambda, podría crear una nueva construcción de L3:

```
export class MyCompanyLambdaFunction extends Function {
  constructor(
    scope: Construct,
    id: string,
    props: Partial<FunctionProps> = {}
  ) {
    super(scope, id, {
      handler: 'index.handler',
      runtime: Runtime.NODEJS_LATEST,
      code: Code.fromAsset(`src/lambda/${props.functionName || id}`),
      ...props
    });
  }
}
```

A continuación, puede reemplazar la `Function` construcción L2 en todas partes del repositorio de la siguiente manera:

```
new MyCompanyLambdaFunction(this, "MyFunction");
new MyCompanyLambdaFunction(this, "MyOtherFunction");
new MyCompanyLambdaFunction(this, "MyThirdFunction", {
  runtime: Runtime.PYTHON_3_11
});
```

Los valores predeterminados le permiten crear nuevas funciones Lambda en una sola línea, y la construcción L3 está configurada para que pueda anular las propiedades predeterminadas si es necesario.

La extensión directa de las construcciones de L2 funciona mejor cuando solo se desean añadir nuevos valores por defecto a las construcciones de L2 existentes. Si también necesitas otra lógica personalizada, es mejor ampliar la clase. Construct La razón de esto se debe al `super` método, al que se invoca desde el constructor. En las clases que amplían otras clases, el `super` método se utiliza para llamar al constructor de la clase principal, y esto debe ser lo primero que ocurra en el constructor. Esto significa que cualquier manipulación de los argumentos pasados u otra lógica personalizada solo puede producirse después de que se haya creado la construcción L2 original. [Si necesitas realizar alguna de estas lógicas personalizadas antes de crear una instancia de la construcción de L2, es mejor seguir el patrón descrito anteriormente en la sección Interacciones entre recursos.](#)

## Recursos personalizados

Los [recursos personalizados](#) son una potente función CloudFormation que permite ejecutar una lógica personalizada desde una función de Lambda que se activa durante el despliegue de la pila. Siempre que necesite algún proceso durante la implementación que no sea compatible directamente con él CloudFormation, puede utilizar un recurso personalizado para hacerlo realidad. AWS CDK Ofrece clases que también le permiten crear recursos personalizados mediante programación. Al usar recursos personalizados dentro de un constructor L3, puede hacer una construcción a partir de casi cualquier cosa.

Una de las ventajas de usar Amazon CloudFront es su sólida capacidad de almacenamiento en caché global. Si quieres restablecer manualmente esa caché para que tu sitio web refleje inmediatamente los nuevos cambios realizados en tu origen, puedes utilizar una [CloudFront invalidación](#). Sin embargo, las invalidaciones son procesos que se ejecutan en una CloudFront distribución en lugar de ser propiedades de una distribución. CloudFront Se pueden crear y aplicar a una distribución existente en cualquier momento, por lo que no forman parte de forma nativa del proceso de aprovisionamiento e implementación.

En este escenario, es posible que desees crear y ejecutar una invalidación después de cada actualización del origen de una distribución. Gracias a los recursos personalizados, puedes crear una construcción L3 similar a la siguiente:

```
export interface CloudFrontInvalidationProps {
```

```

    distribution: Distribution
    region?: string
    paths?: string[]
  }

export class CloudFrontInvalidation extends Construct {
  constructor(
    scope: Construct,
    id: string,
    props: CloudFrontInvalidationProps
  ) {
    super(scope, id);
    const policy = AwsCustomResourcePolicy.fromSdkCalls({
      resources: AwsCustomResourcePolicy.ANY_RESOURCE
    });
    new AwsCustomResource(scope, `${id}Invalidation`, {
      policy,
      onUpdate: {
        service: 'CloudFront',
        action: 'createInvalidation',
        region: props.region || 'us-east-1',
        physicalResourceId:
PhysicalResourceId.fromResponse('Invalidation.Id'),
        parameters: {
          DistributionId: props.distribution.distributionId,
          InvalidationBatch: {
            Paths: {
              Quantity: props.paths?.length || 1,
              Items: props.paths || ['/*']
            },
            CallerReference: crypto.randomBytes(5).toString('hex')
          }
        }
      }
    });
  }
}

```

Con la distribución que creamos anteriormente en la construcción `CloudFrontWebsite L3`, podrías hacerlo muy fácilmente:

```

new CloudFrontInvalidation(this, 'MyInvalidation', {
  distribution: siteADotCom.distribution

```

```
});
```

Esta construcción L3 usa una construcción AWS CDK L3 llamada [AwsCustomResource](#) para crear un recurso personalizado que ejecute una lógica personalizada. `AwsCustomResource` es muy práctico cuando necesita realizar exactamente una llamada al SDK de AWS, ya que le permite hacerlo sin tener que escribir ningún código Lambda. Si tiene requisitos más complejos y quiere implementar su propia lógica, puede utilizar directamente la [CustomResource](#) clase básica.

Otro buen ejemplo del AWS CDK uso de una construcción L3 de recursos personalizada es el [despliegue de cubos de S3](#). La función Lambda creada por el recurso personalizado dentro del constructor de esta construcción de L3 añade funcionalidades que CloudFormation no se podrían gestionar de otro modo: añade y actualiza objetos en un bucket de S3. Sin la implementación de un bucket de S3, no podrías colocar contenido en el bucket de S3 que acabas de crear como parte de tu stack, lo que sería muy inconveniente.

El mejor ejemplo de cómo AWS CDK eliminar la necesidad de escribir montones de CloudFormation sintaxis es este básico: `S3BucketDeployment`

```
new BucketDeployment(this, 'BucketObjects', {
  sources: [Source.asset('./path/to/amzn-s3-demo-bucket')],
  destinationBucket: amzn-s3-demo-bucket
});
```

Compáralo con el CloudFormation código que tendrías que escribir para lograr lo mismo:

```
"lambdapolicyA5E98E09": {
  "Type": "AWS::IAM::Policy",
  "Properties": {
    "PolicyDocument": {
      "Statement": [
        {
          "Action": "lambda:UpdateFunctionCode",
          "Effect": "Allow",
          "Resource": "arn:aws:lambda:us-east-1:123456789012:function:my-function"
        }
      ],
      "Version": "2012-10-17"
    },
    "PolicyName": "lambdaPolicy",
    "Roles": [
      {
```

```

    "Ref": "myiamroleF09C7974"
  }
]
},
"Metadata": {
  "aws:cdk:path": "CdkScratchStack/lambda-policy/Resource"
}
},
"BucketObjectsAwsCliLayer8C081206": {
  "Type": "AWS::Lambda::LayerVersion",
  "Properties": {
    "Content": {
      "S3Bucket": {
        "Fn::Sub": "cdk-hnb659fds-assets-${AWS::AccountId}-${AWS::Region}"
      },
      "S3Key": "e2277687077a2abf9ae1af1cc9565e6715e2ebb62f79ec53aa75a1af9298f642.zip"
    },
    "Description": "/opt/awscli/aws"
  },
  "Metadata": {
    "aws:cdk:path": "CdkScratchStack/BucketObjects/AwsCliLayer/Resource",
    "aws:asset:path":
"asset.e2277687077a2abf9ae1af1cc9565e6715e2ebb62f79ec53aa75a1af9298f642.zip",
    "aws:asset:is-bundled": false,
    "aws:asset:property": "Content"
  }
},
"BucketObjectsCustomResourceB12E6837": {
  "Type": "Custom::CDKBucketDeployment",
  "Properties": {
    "ServiceToken": {
      "Fn::GetAtt": [
        "CustomCDKBucketDeployment8693BB64968944B69Aafb0CC9EB8756C81C01536",
        "Arn"
      ]
    }
  },
  "SourceBucketNames": [
    {
      "Fn::Sub": "cdk-hnb659fds-assets-${AWS::AccountId}-${AWS::Region}"
    }
  ],
  "SourceObjectKeys": [
    "f888a9d977f0b5bdbc04a1f8f07520ede6e00d4051b9a6a250860a1700924f26.zip"
  ],

```

```

    "DestinationBucketName": {
      "Ref": "amzn-s3-demo-bucket77F80CC0"
    },
    "Prune": true
  },
  "UpdateReplacePolicy": "Delete",
  "DeletionPolicy": "Delete",
  "Metadata": {
    "aws:cdk:path": "CdkScratchStack/BucketObjects/CustomResource/Default"
  }
},
"CustomCDKBucketDeployment8693BB64968944B69AAFB0CC9EB8756CServiceRole89A01265": {
  "Type": "AWS::IAM::Role",
  "Properties": {
    "AssumeRolePolicyDocument": {
      "Statement": [
        {
          "Action": "sts:AssumeRole",
          "Effect": "Allow",
          "Principal": {
            "Service": "lambda.amazonaws.com"
          }
        }
      ],
      "Version": "2012-10-17"
    },
    "ManagedPolicyArns": [
      {
        "Fn::Join": [
          "",
          [
            "arn:",
            {
              "Ref": "AWS::Partition"
            },
            ":iam::aws:policy/service-role/AWSLambdaBasicExecutionRole"
          ]
        ]
      }
    ]
  },
  "Metadata": {
    "aws:cdk:path": "CdkScratchStack/
Custom::CDKBucketDeployment8693BB64968944B69AAFB0CC9EB8756C/ServiceRole/Resource"

```

```

    }
  },

  "CustomCDKBucketDeployment8693BB64968944B69Aafb0cc9EB8756CServiceRoleDefaultPolicy88902FDF":
  {
    "Type": "AWS::IAM::Policy",
    "Properties": {
      "PolicyDocument": {
        "Statement": [
          {
            "Action": [
              "s3:GetBucket*",
              "s3:GetObject*",
              "s3:List*"
            ],
            "Effect": "Allow",
            "Resource": [
              {
                "Fn::Join": [
                  "",
                  [
                    "arn:",
                    {
                      "Ref": "AWS::Partition"
                    },
                    ":s3::",
                    {
                      "Fn::Sub": "cdk-hnb659fds-assets-${AWS::AccountId}-${AWS::Region}"
                    },
                    "/*"
                  ]
                ]
              }
            ],
          },
          {
            "Fn::Join": [
              "",
              [
                "arn:",
                {
                  "Ref": "AWS::Partition"
                },
                ":s3::",
                {
                  "Fn::Sub": "cdk-hnb659fds-assets-${AWS::AccountId}-${AWS::Region}"
                }
              ]
            ]
          }
        ]
      }
    }
  }
}

```

```
    }
  ]
]
}
]
},
{
  "Action": [
    "s3:Abort*",
    "s3:DeleteObject*",
    "s3:GetBucket*",
    "s3:GetObject*",
    "s3:List*",
    "s3:PutObject",
    "s3:PutObjectLegalHold",
    "s3:PutObjectRetention",
    "s3:PutObjectTagging",
    "s3:PutObjectVersionTagging"
  ],
  "Effect": "Allow",
  "Resource": [
    {
      "Fn::GetAtt": [
        "amzn3demobucket77F80CC0",
        "Arn"
      ]
    },
    {
      "Fn::Join": [
        "",
        [
          {
            "Fn::GetAtt": [
              "amzn3demobucket77F80CC0",
              "Arn"
            ]
          },
          "/"
        ]
      ]
    }
  ]
}
],
```

```

    "Version": "2012-10-17"
  },
  "PolicyName":
"CustomCDKBucketDeployment8693BB64968944B69Aafb0cc9EB8756CServiceRoleDefaultPolicy88902FDF",
  "Roles": [
    {
      "Ref":
"CustomCDKBucketDeployment8693BB64968944B69Aafb0cc9EB8756CServiceRole89A01265"
    }
  ],
  "Metadata": {
    "aws:cdk:path": "CdkScratchStack/
Custom::CDKBucketDeployment8693BB64968944B69Aafb0cc9EB8756C/ServiceRole/DefaultPolicy/
Resource"
  }
},
"CustomCDKBucketDeployment8693BB64968944B69Aafb0cc9EB8756C81C01536": {
  "Type": "AWS::Lambda::Function",
  "Properties": {
    "Code": {
      "S3Bucket": {
        "Fn::Sub": "cdk-hnb659fds-assets-${AWS::AccountId}-${AWS::Region}"
      },
      "S3Key": "9eb41a5505d37607ac419321497a4f8c21cf0ee1f9b4a6b29aa04301aea5c7fd.zip"
    },
    "Role": {
      "Fn::GetAtt": [
        "CustomCDKBucketDeployment8693BB64968944B69Aafb0cc9EB8756CServiceRole89A01265",
        "Arn"
      ]
    },
    "Environment": {
      "Variables": {
        "AWS_CA_BUNDLE": "/etc/pki/ca-trust/extracted/pem/tls-ca-bundle.pem"
      }
    },
    "Handler": "index.handler",
    "Layers": [
      {
        "Ref": "BucketObjectsAwsCliLayer8C081206"
      }
    ],
    "Runtime": "python3.9",

```

```
    "Timeout": 900
  },
  "DependsOn": [

    "CustomCDKBucketDeployment8693BB64968944B69Aafb0cc9EB8756CServiceRoleDefaultPolicy88902FDF",
    "CustomCDKBucketDeployment8693BB64968944B69Aafb0cc9EB8756CServiceRole89A01265"
  ],
  "Metadata": {
    "aws:cdk:path": "CdkScratchStack/
Custom::CDKBucketDeployment8693BB64968944B69Aafb0cc9EB8756C/Resource",
    "aws:asset:path":
"asset.9eb41a5505d37607ac419321497a4f8c21cf0ee1f9b4a6b29aa04301aea5c7fd",
    "aws:asset:is-bundled": false,
    "aws:asset:property": "Code"
  }
}
```

¡4 líneas contra 241 líneas es una gran diferencia! Y esto es solo un ejemplo de lo que es posible cuando aprovechas la capa 3 para personalizar tus pilas.

# Prácticas recomendadas

## La capa 1 construye

- No siempre se puede evitar el uso directo de las construcciones de L1, pero hay que evitarlas siempre que sea posible. Si una construcción L2 específica no es compatible con tu hipótesis límite, puedes explorar estas dos opciones en lugar de usar la construcción L1 directamente:
  - `AccessDefaultChild`: si la CloudFormation propiedad que necesita no está disponible en una construcción de L2, puede acceder a la construcción de L1 subyacente utilizando `L2Construct.node.defaultChild`. Puede actualizar cualquier propiedad pública de la construcción L1 accediendo a ella a través de esta propiedad, en lugar de tomarse la molestia de crear la construcción L1 usted mismo.
  - Utilice anulaciones de propiedades: ¿Qué sucede si la propiedad que desea actualizar no es pública? La mejor trampilla de escape que permite AWS CDK hacer todo lo que puede hacer una CloudFormation plantilla es utilizar un método que está disponible en todas las construcciones de L1: [addPropertyOverride](#). Puedes manipular tu pila a nivel de CloudFormation plantilla pasando el nombre y el valor de la CloudFormation propiedad directamente a este método.

## Construcciones L2

- Recuerde aprovechar los métodos auxiliares que suelen ofrecer las construcciones L2. Con la capa 2, no es necesario pasar todas las propiedades al instanciar. Los métodos auxiliares de nivel 2 pueden hacer que el aprovisionamiento de recursos sea exponencialmente más práctico, especialmente cuando se necesita lógica condicional. [Uno de los métodos de ayuda más convenientes se deriva de la clase Grant](#). Esta clase no se usa directamente, pero muchas construcciones de nivel 2 la utilizan para proporcionar métodos auxiliares que facilitan mucho la implementación de los permisos. Por ejemplo, si quiere dar permiso a una función Lambda de nivel 2 para acceder a un bucket de nivel 2 de S3, puede `s3Bucket.grantReadWrite(lambdaFunction)` llamar a una nueva función y política en lugar de crear.

## Construcciones de nivel 3

- Si bien las construcciones L3 pueden resultar muy prácticas cuando quieres que tus pilas sean más reutilizables y personalizables, te recomendamos que las utilices con cuidado. Considera qué tipo de construcción L3 necesitas o si necesitas alguna construcción L3:
  - Si no interactúas directamente con AWS los recursos, suele ser más apropiado crear una clase auxiliar en lugar de ampliarla. `Construct` Esto se debe a que la `Construct` clase realiza muchas acciones de forma predeterminada que solo son necesarias si interactúas directamente con AWS los recursos. Por lo tanto, si no necesitas que se realicen esas acciones, es más eficiente evitarlas.
  - Si determina que es apropiado crear una nueva construcción de L3, en la mayoría de los casos querrá ampliar la `Construct` clase directamente. Extienda otras construcciones de L2 solo cuando desee actualizar las propiedades por defecto de la construcción. Si están involucradas otras construcciones de L2 o una lógica personalizada, extienda `Construct` directamente e instancie todos los recursos del constructor.

## Preguntas frecuentes

### ¿No puedo usar las capas AWS CDK sin entender las capas?

No cabe duda de que sí. Pero como ocurre con las herramientas más poderosas, AWS CDK se vuelven más poderosas cuanto más sabes sobre ellas. Al aprender cómo interactúan las capas, se obtiene un nuevo nivel de comprensión que ayuda a simplificar las implementaciones de pilas mucho más allá de lo que se puede hacer con solo unos conocimientos básicos AWS CDK . AWS CDK

### ¿Puedo crear construcciones de L2 a partir de L1 de la misma manera que hago construcciones de L3 a partir de L2?

Si un recurso ya tiene una construcción de L2, le recomendamos que utilice esa construcción y realice las personalizaciones en la capa 3. Esto se debe a que ya se han realizado muchas investigaciones para determinar las mejores formas de configurar las construcciones de L2 existentes para un recurso en particular. Sin embargo, hay varias construcciones de L1 cuyas construcciones de L2 aún no existen. En esos casos, le animamos a crear sus propias construcciones de L2 y a compartirlas con otras personas, convirtiéndose en colaborador de la biblioteca de código abierto. AWS CDK Encontrará todo lo que necesita para empezar en [las directrices de contribución](#) de la. AWS CDK

### ¿Qué AWS recursos aún no tienen construcciones oficiales de L2?

[La cantidad de AWS recursos que no tienen construcciones de L2 disminuye día a día, pero si estás interesado en ayudar a crear una construcción de L2 para uno de estos recursos, consulta la referencia de la API.AWS CDK](#) Consulta la lista de recursos en el panel izquierdo. Los recursos que tienen el superíndice 1 junto a sus nombres no tienen construcciones L2 oficiales.

### ¿Puedo hacer una construcción de L2 o L3 en cualquier idioma que sea compatible? AWS CDK

AWS CDK Es compatible con varios lenguajes de programación TypeScript JavaScript, incluidos Python, Java, C# y Go. Puede crear sus construcciones L3 personales utilizando el AWS CDK código compilado en el lenguaje correspondiente. Sin embargo, si desea contribuir a las AWS

CDK construcciones nativas AWS CDK o crearlas, debe utilizarlas. TypeScript Esto se debe a que TypeScript es el único idioma nativo de AWS CDK. Las AWS CDK versiones para otros idiomas se crean a partir del TypeScript código nativo mediante una AWS biblioteca llamada [JSii](#).

## ¿Dónde puedo encontrar las construcciones de L3 existentes fuera de? AWS CDK

[Hay demasiadas ubicaciones para compartirlas aquí, pero puede encontrar muchas de las construcciones más populares en el sitio web de AWS Solutions Constructs y en la AWS CDK sección del Construct Hub.](#)

# Recursos

- [AWS CDK Referencia de la API](#)
- [AWS CloudFormation especificación de recursos](#)
- [AWS CDK construye la documentación](#)
- [AWS CDK abstracciones y trampillas de escape](#)
- [Aproveche las construcciones L2 para reducir la complejidad de su AWS CDK aplicación \(entrada de blog\)AWS](#)
- [AWS CloudFormation recursos personalizados](#)
- [AWS Construcciones de soluciones](#)
- [Construye un centro](#)
- [AWS CDK Ejemplos](#) (GitHub repositorio)

## Historial de documentos

En la siguiente tabla, se describen cambios significativos de esta guía. Si quiere recibir notificaciones de futuras actualizaciones, puede suscribirse a las [notificaciones RSS](#).

Cambio	Descripción	Fecha
<a href="#">Publicación inicial</a>	—	4 de diciembre de 2023

# AWS Glosario de orientación prescriptiva

Los siguientes son términos de uso común en las estrategias, guías y patrones proporcionados por la Guía AWS prescriptiva. Para sugerir entradas, utilice el enlace [Enviar comentarios](#) al final del glosario.

## Números

### Las 7 R

Siete estrategias de migración comunes para trasladar aplicaciones a la nube. Estas estrategias se basan en las 5 R que Gartner identificó en 2011 y consisten en lo siguiente:

- **Refactorizar/rediseñar:** traslade una aplicación y modifique su arquitectura mediante el máximo aprovechamiento de las características nativas en la nube para mejorar la agilidad, el rendimiento y la escalabilidad. Por lo general, esto implica trasladar el sistema operativo y la base de datos. Ejemplo: migre su base de datos Oracle local a la edición compatible con PostgreSQL de Amazon Aurora.
- **Redefinir la plataforma (transportar y redefinir):** traslade una aplicación a la nube e introduzca algún nivel de optimización para aprovechar las capacidades de la nube. Ejemplo: migre su base de datos Oracle local a Amazon Relational Database Service (Amazon RDS) para Oracle en el Nube de AWS
- **Recomprar (readquirir):** cambie a un producto diferente, lo cual se suele llevar a cabo al pasar de una licencia tradicional a un modelo SaaS. Ejemplo: migre su sistema de gestión de relaciones con los clientes (CRM) a Salesforce.com.
- **Volver a alojar (migrar mediante lift-and-shift):** traslade una aplicación a la nube sin realizar cambios para aprovechar las capacidades de la nube. Ejemplo: migre su base de datos Oracle local a Oracle en una EC2 instancia del Nube de AWS
- **Reubicar:** (migrar el hipervisor mediante lift and shift): traslade la infraestructura a la nube sin comprar equipo nuevo, reescribir aplicaciones o modificar las operaciones actuales. Los servidores se migran de una plataforma local a un servicio en la nube para la misma plataforma. Ejemplo: migrar una Microsoft Hyper-V aplicación a AWS.
- **Retener (revisitar):** conserve las aplicaciones en el entorno de origen. Estas pueden incluir las aplicaciones que requieren una refactorización importante, que desee posponer para más adelante, y las aplicaciones heredadas que desee retener, ya que no hay ninguna justificación empresarial para migrarlas.

- Retirar: retire o elimine las aplicaciones que ya no sean necesarias en un entorno de origen.

## A

### ABAC

Consulte control de [acceso basado en atributos](#).

### servicios abstractos

Consulte [servicios gestionados](#).

### ACID

Consulte [atomicidad, consistencia, aislamiento y durabilidad](#).

### migración activa-activa

Método de migración de bases de datos en el que las bases de datos de origen y destino se mantienen sincronizadas (mediante una herramienta de replicación bidireccional o mediante operaciones de escritura doble) y ambas bases de datos gestionan las transacciones de las aplicaciones conectadas durante la migración. Este método permite la migración en lotes pequeños y controlados, en lugar de requerir una transición única. Es más flexible, pero requiere más trabajo que la migración [activa-pasiva](#).

### migración activa-pasiva

Método de migración de bases de datos en el que las bases de datos de origen y destino se mantienen sincronizadas, pero solo la base de datos de origen gestiona las transacciones de las aplicaciones conectadas mientras los datos se replican en la base de datos de destino. La base de datos de destino no acepta ninguna transacción durante la migración.

### función de agregación

Función SQL que opera en un grupo de filas y calcula un único valor de retorno para el grupo. Entre los ejemplos de funciones agregadas se incluyen SUM y MAX.

### IA

Véase [inteligencia artificial](#).

### AIOps

Consulte las [operaciones de inteligencia artificial](#).

## anonimización

El proceso de eliminar permanentemente la información personal de un conjunto de datos. La anonimización puede ayudar a proteger la privacidad personal. Los datos anonimizados ya no se consideran datos personales.

## antipatrones

Una solución que se utiliza con frecuencia para un problema recurrente en el que la solución es contraproducente, ineficaz o menos eficaz que una alternativa.

## control de aplicaciones

Un enfoque de seguridad que permite el uso únicamente de aplicaciones aprobadas para ayudar a proteger un sistema contra el malware.

## cartera de aplicaciones

Recopilación de información detallada sobre cada aplicación que utiliza una organización, incluido el costo de creación y mantenimiento de la aplicación y su valor empresarial. Esta información es clave para [el proceso de detección y análisis de la cartera](#) y ayuda a identificar y priorizar las aplicaciones que se van a migrar, modernizar y optimizar.

## inteligencia artificial (IA)

El campo de la informática que se dedica al uso de tecnologías informáticas para realizar funciones cognitivas que suelen estar asociadas a los seres humanos, como el aprendizaje, la resolución de problemas y el reconocimiento de patrones. Para más información, consulte [¿Qué es la inteligencia artificial?](#)

## operaciones de inteligencia artificial (AIOps)

El proceso de utilizar técnicas de machine learning para resolver problemas operativos, reducir los incidentes operativos y la intervención humana, y mejorar la calidad del servicio. Para obtener más información sobre cómo AIOps se utiliza en la estrategia de AWS migración, consulte la [guía de integración de operaciones](#).

## cifrado asimétrico

Algoritmo de cifrado que utiliza un par de claves, una clave pública para el cifrado y una clave privada para el descifrado. Puede compartir la clave pública porque no se utiliza para el descifrado, pero el acceso a la clave privada debe estar sumamente restringido.

## atomicidad, consistencia, aislamiento, durabilidad (ACID)

Conjunto de propiedades de software que garantizan la validez de los datos y la fiabilidad operativa de una base de datos, incluso en caso de errores, cortes de energía u otros problemas.

## control de acceso basado en atributos (ABAC)

La práctica de crear permisos detallados basados en los atributos del usuario, como el departamento, el puesto de trabajo y el nombre del equipo. Para obtener más información, consulte [ABAC AWS en la](#) documentación AWS Identity and Access Management (IAM).

## origen de datos fidedigno

Ubicación en la que se almacena la versión principal de los datos, que se considera la fuente de información más fiable. Puede copiar los datos del origen de datos autorizado a otras ubicaciones con el fin de procesarlos o modificarlos, por ejemplo, anonimizarlos, redactarlos o seudonimizarlos.

## Zona de disponibilidad

Una ubicación distinta dentro de una Región de AWS que está aislada de los fallos en otras zonas de disponibilidad y que proporciona una conectividad de red económica y de baja latencia a otras zonas de disponibilidad de la misma región.

## AWS Marco de adopción de la nube (AWS CAF)

Un marco de directrices y mejores prácticas AWS para ayudar a las organizaciones a desarrollar un plan eficiente y eficaz para migrar con éxito a la nube. AWS CAF organiza la orientación en seis áreas de enfoque denominadas perspectivas: negocios, personas, gobierno, plataforma, seguridad y operaciones. Las perspectivas empresariales, humanas y de gobernanza se centran en las habilidades y los procesos empresariales; las perspectivas de plataforma, seguridad y operaciones se centran en las habilidades y los procesos técnicos. Por ejemplo, la perspectiva humana se dirige a las partes interesadas que se ocupan de los Recursos Humanos (RR. HH.), las funciones del personal y la administración de las personas. Desde esta perspectiva, AWS CAF proporciona orientación para el desarrollo, la formación y la comunicación de las personas a fin de preparar a la organización para una adopción exitosa de la nube. Para obtener más información, consulte la [Página web de AWS CAF](#) y el [Documento técnico de AWS CAF](#).

## AWS Marco de calificación de la carga de trabajo (AWS WQF)

Herramienta que evalúa las cargas de trabajo de migración de bases de datos, recomienda estrategias de migración y proporciona estimaciones de trabajo. AWS WQF se incluye con AWS

Schema Conversion Tool ().AWS SCT Analiza los esquemas de bases de datos y los objetos de código, el código de las aplicaciones, las dependencias y las características de rendimiento y proporciona informes de evaluación.

## B

Un bot malo

Un [bot](#) destinado a interrumpir o causar daño a personas u organizaciones.

BCP

Consulte la [planificación de la continuidad del negocio](#).

gráfico de comportamiento

Una vista unificada e interactiva del comportamiento de los recursos y de las interacciones a lo largo del tiempo. Puede utilizar un gráfico de comportamiento con Amazon Detective para examinar los intentos de inicio de sesión fallidos, las llamadas sospechosas a la API y acciones similares. Para obtener más información, consulte [Datos en un gráfico de comportamiento](#) en la documentación de Detective.

sistema big-endian

Un sistema que almacena primero el byte más significativo. Véase también [endianness](#).

clasificación binaria

Un proceso que predice un resultado binario (una de las dos clases posibles). Por ejemplo, es posible que su modelo de ML necesite predecir problemas como “¿Este correo electrónico es spam o no es spam?” o “¿Este producto es un libro o un automóvil?”.

filtro de floración

Estructura de datos probabilística y eficiente en términos de memoria que se utiliza para comprobar si un elemento es miembro de un conjunto.

implementación azul/verde

Una estrategia de despliegue en la que se crean dos entornos separados pero idénticos. La versión actual de la aplicación se ejecuta en un entorno (azul) y la nueva versión de la aplicación en el otro entorno (verde). Esta estrategia le ayuda a revertirla rápidamente con un impacto mínimo.

## bot

Una aplicación de software que ejecuta tareas automatizadas a través de Internet y simula la actividad o interacción humana. Algunos bots son útiles o beneficiosos, como los rastreadores web que indexan información en Internet. Algunos otros bots, conocidos como bots malos, tienen como objetivo interrumpir o causar daños a personas u organizaciones.

## botnet

Redes de [bots](#) que están infectadas por [malware](#) y que están bajo el control de una sola parte, conocida como pastor u operador de bots. Las botnets son el mecanismo más conocido para escalar los bots y su impacto.

## branch

Área contenida de un repositorio de código. La primera rama que se crea en un repositorio es la rama principal. Puede crear una rama nueva a partir de una rama existente y, a continuación, desarrollar características o corregir errores en la rama nueva. Una rama que se genera para crear una característica se denomina comúnmente rama de característica. Cuando la característica se encuentra lista para su lanzamiento, se vuelve a combinar la rama de característica con la rama principal. Para obtener más información, consulte [Acerca de las sucursales](#) (GitHub documentación).

## acceso con cristales rotos

En circunstancias excepcionales y mediante un proceso aprobado, un usuario puede acceder rápidamente a un sitio para el Cuenta de AWS que normalmente no tiene permisos de acceso. Para obtener más información, consulte el indicador [Implemente procedimientos de rotura de cristales en la guía Well-Architected AWS](#) .

## estrategia de implementación sobre infraestructura existente

La infraestructura existente en su entorno. Al adoptar una estrategia de implementación sobre infraestructura existente para una arquitectura de sistemas, se diseña la arquitectura en función de las limitaciones de los sistemas y la infraestructura actuales. Si está ampliando la infraestructura existente, puede combinar las estrategias de implementación sobre infraestructuras existentes y de [implementación desde cero](#).

## caché de búfer

El área de memoria donde se almacenan los datos a los que se accede con más frecuencia.

## capacidad empresarial

Lo que hace una empresa para generar valor (por ejemplo, ventas, servicio al cliente o marketing). Las arquitecturas de microservicios y las decisiones de desarrollo pueden estar impulsadas por las capacidades empresariales. Para obtener más información, consulte la sección [Organizado en torno a las capacidades empresariales](#) del documento técnico [Ejecutar microservicios en contenedores en AWS](#).

## planificación de la continuidad del negocio (BCP)

Plan que aborda el posible impacto de un evento disruptivo, como una migración a gran escala en las operaciones y permite a la empresa reanudar las operaciones rápidamente.

# C

## CAF

[Consulte el marco AWS de adopción de la nube.](#)

## despliegue canario

El lanzamiento lento e incremental de una versión para los usuarios finales. Cuando se tiene confianza, se despliega la nueva versión y se reemplaza la versión actual en su totalidad.

## CCoE

Consulte [Cloud Center of Excellence](#).

## CDC

Consulte la [captura de datos de cambios](#).

## captura de datos de cambio (CDC)

Proceso de seguimiento de los cambios en un origen de datos, como una tabla de base de datos, y registro de los metadatos relacionados con el cambio. Puede utilizar los CDC para diversos fines, como auditar o replicar los cambios en un sistema de destino para mantener la sincronización.

## ingeniería del caos

Introducir intencionalmente fallos o eventos disruptivos para poner a prueba la resiliencia de un sistema. Puedes usar [AWS Fault Injection Service \(AWS FIS\)](#) para realizar experimentos que estresen tus AWS cargas de trabajo y evalúen su respuesta.

## CI/CD

Consulte la [integración continua y la entrega continua](#).

### clasificación

Un proceso de categorización que permite generar predicciones. Los modelos de ML para problemas de clasificación predicen un valor discreto. Los valores discretos siempre son distintos entre sí. Por ejemplo, es posible que un modelo necesite evaluar si hay o no un automóvil en una imagen.

### cifrado del cliente

Cifrado de datos localmente, antes de que el objetivo los Servicio de AWS reciba.

### Centro de excelencia en la nube (CCoE)

Equipo multidisciplinario que impulsa los esfuerzos de adopción de la nube en toda la organización, incluido el desarrollo de las prácticas recomendadas en la nube, la movilización de recursos, el establecimiento de plazos de migración y la dirección de la organización durante las transformaciones a gran escala. Para obtener más información, consulte las [publicaciones de CCoE](#) en el blog de estrategia Nube de AWS empresarial.

### computación en la nube

La tecnología en la nube que se utiliza normalmente para la administración de dispositivos de IoT y el almacenamiento de datos de forma remota. La computación en la nube suele estar conectada a la tecnología de [computación perimetral](#).

### modelo operativo en la nube

En una organización de TI, el modelo operativo que se utiliza para crear, madurar y optimizar uno o más entornos de nube. Para obtener más información, consulte [Creación de su modelo operativo de nube](#).

### etapas de adopción de la nube

Las cuatro fases por las que suelen pasar las organizaciones cuando migran a Nube de AWS:

- Proyecto: ejecución de algunos proyectos relacionados con la nube con fines de prueba de concepto y aprendizaje
- Fundamento: realizar inversiones fundamentales para escalar su adopción de la nube (p. ej., crear una landing zone, definir una CCoE, establecer un modelo de operaciones)
- Migración: migración de aplicaciones individuales
- Reinención: optimización de productos y servicios e innovación en la nube

Stephen Orban definió estas etapas en la entrada del blog [The Journey Toward Cloud-First & the Stages of Adoption en el](#) blog Nube de AWS Enterprise Strategy. Para obtener información sobre su relación con la estrategia de AWS migración, consulte la guía de [preparación para la migración](#).

## CMDB

Consulte la [base de datos de administración de la configuración](#).

## repositorio de código

Una ubicación donde el código fuente y otros activos, como documentación, muestras y scripts, se almacenan y actualizan mediante procesos de control de versiones. Los repositorios en la nube más comunes incluyen GitHub o Bitbucket Cloud. Cada versión del código se denomina rama. En una estructura de microservicios, cada repositorio se encuentra dedicado a una única funcionalidad. Una sola canalización de CI/CD puede utilizar varios repositorios.

## caché en frío

Una caché de búfer que está vacía no está bien poblada o contiene datos obsoletos o irrelevantes. Esto afecta al rendimiento, ya que la instancia de la base de datos debe leer desde la memoria principal o el disco, lo que es más lento que leer desde la memoria caché del búfer.

## datos fríos

Datos a los que se accede con poca frecuencia y que suelen ser históricos. Al consultar este tipo de datos, normalmente se aceptan consultas lentas. Trasladar estos datos a niveles o clases de almacenamiento de menor rendimiento y menos costosos puede reducir los costos.

## visión artificial (CV)

Campo de la [IA](#) que utiliza el aprendizaje automático para analizar y extraer información de formatos visuales, como imágenes y vídeos digitales. Por ejemplo, Amazon SageMaker AI proporciona algoritmos de procesamiento de imágenes para CV.

## desviación de configuración

En el caso de una carga de trabajo, un cambio de configuración con respecto al estado esperado. Puede provocar que la carga de trabajo deje de cumplir las normas y, por lo general, es gradual e involuntario.

## base de datos de administración de configuración (CMDB)

Repositorio que almacena y administra información sobre una base de datos y su entorno de TI, incluidos los componentes de hardware y software y sus configuraciones. Por lo general, los

datos de una CMDB se utilizan en la etapa de detección y análisis de la cartera de productos durante la migración.

## paquete de conformidad

Conjunto de AWS Config reglas y medidas correctivas que puede reunir para personalizar sus comprobaciones de conformidad y seguridad. Puede implementar un paquete de conformidad como una entidad única en una región Cuenta de AWS y, o en una organización, mediante una plantilla YAML. Para obtener más información, consulta los [paquetes de conformidad](#) en la documentación. AWS Config

## integración y entrega continuas (CI/CD)

El proceso de automatización de las etapas de origen, compilación, prueba, puesta en escena y producción del proceso de publicación del software. CI/CD se describe comúnmente como una canalización. CI/CD puede ayudarlo a automatizar los procesos, mejorar la productividad, mejorar la calidad del código y entregar más rápido. Para obtener más información, consulte [Beneficios de la entrega continua](#). CD también puede significar implementación continua. Para obtener más información, consulte [Entrega continua frente a implementación continua](#).

## CV

Vea la [visión artificial](#).

## D

### datos en reposo

Datos que están estacionarios en la red, como los datos que se encuentran almacenados.

### clasificación de datos

Un proceso para identificar y clasificar los datos de su red en función de su importancia y sensibilidad. Es un componente fundamental de cualquier estrategia de administración de riesgos de ciberseguridad porque lo ayuda a determinar los controles de protección y retención adecuados para los datos. La clasificación de datos es un componente del pilar de seguridad del AWS Well-Architected Framework. Para obtener más información, consulte [Clasificación de datos](#).

### desviación de datos

Una variación significativa entre los datos de producción y los datos que se utilizaron para entrenar un modelo de machine learning, o un cambio significativo en los datos de entrada

a lo largo del tiempo. La desviación de los datos puede reducir la calidad, la precisión y la imparcialidad generales de las predicciones de los modelos de machine learning.

#### datos en tránsito

Datos que se mueven de forma activa por la red, por ejemplo, entre los recursos de la red.

#### mallado de datos

Un marco arquitectónico que proporciona una propiedad de datos distribuida y descentralizada con administración y gobierno centralizados.

#### minimización de datos

El principio de recopilar y procesar solo los datos estrictamente necesarios. Practicar la minimización de los datos Nube de AWS puede reducir los riesgos de privacidad, los costos y la huella de carbono de la analítica.

#### perímetro de datos

Un conjunto de barreras preventivas en su AWS entorno que ayudan a garantizar que solo las identidades confiables accedan a los recursos confiables desde las redes esperadas. Para obtener más información, consulte [Crear un perímetro de datos sobre](#). AWS

#### preprocesamiento de datos

Transformar los datos sin procesar en un formato que su modelo de ML pueda analizar fácilmente. El preprocesamiento de datos puede implicar eliminar determinadas columnas o filas y corregir los valores faltantes, incoherentes o duplicados.

#### procedencia de los datos

El proceso de rastrear el origen y el historial de los datos a lo largo de su ciclo de vida, por ejemplo, la forma en que se generaron, transmitieron y almacenaron los datos.

#### titular de los datos

Persona cuyos datos se recopilan y procesan.

#### almacenamiento de datos

Un sistema de administración de datos que respalde la inteligencia empresarial, como el análisis. Los almacenes de datos suelen contener grandes cantidades de datos históricos y, por lo general, se utilizan para consultas y análisis.

## lenguaje de definición de datos (DDL)

Instrucciones o comandos para crear o modificar la estructura de tablas y objetos de una base de datos.

## lenguaje de manipulación de datos (DML)

Instrucciones o comandos para modificar (insertar, actualizar y eliminar) la información de una base de datos.

## DDL

Consulte el [lenguaje de definición de bases](#) de datos.

## conjunto profundo

Combinar varios modelos de aprendizaje profundo para la predicción. Puede utilizar conjuntos profundos para obtener una predicción más precisa o para estimar la incertidumbre de las predicciones.

## aprendizaje profundo

Un subcampo del ML que utiliza múltiples capas de redes neuronales artificiales para identificar el mapeo entre los datos de entrada y las variables objetivo de interés.

## defense-in-depth

Un enfoque de seguridad de la información en el que se distribuyen cuidadosamente una serie de mecanismos y controles de seguridad en una red informática para proteger la confidencialidad, la integridad y la disponibilidad de la red y de los datos que contiene. Al adoptar esta estrategia AWS, se añaden varios controles en diferentes capas de la AWS Organizations estructura para ayudar a proteger los recursos. Por ejemplo, un defense-in-depth enfoque podría combinar la autenticación multifactorial, la segmentación de la red y el cifrado.

## administrador delegado

En AWS Organizations, un servicio compatible puede registrar una cuenta de AWS miembro para administrar las cuentas de la organización y gestionar los permisos de ese servicio. Esta cuenta se denomina administrador delegado para ese servicio. Para obtener más información y una lista de servicios compatibles, consulte [Servicios que funcionan con AWS Organizations](#) en la documentación de AWS Organizations .

## Implementación

El proceso de hacer que una aplicación, características nuevas o correcciones de código se encuentren disponibles en el entorno de destino. La implementación abarca implementar

cambios en una base de código y, a continuación, crear y ejecutar esa base en los entornos de la aplicación.

## entorno de desarrollo

Consulte [entorno](#).

## control de detección

Un control de seguridad que se ha diseñado para detectar, registrar y alertar después de que se produzca un evento. Estos controles son una segunda línea de defensa, ya que lo advierten sobre los eventos de seguridad que han eludido los controles preventivos establecidos. Para obtener más información, consulte [Controles de detección](#) en Implementación de controles de seguridad en AWS.

## asignación de flujos de valor para el desarrollo (DVSM)

Proceso que se utiliza para identificar y priorizar las restricciones que afectan negativamente a la velocidad y la calidad en el ciclo de vida del desarrollo de software. DVSM amplía el proceso de asignación del flujo de valor diseñado originalmente para las prácticas de fabricación ajustada. Se centra en los pasos y los equipos necesarios para crear y transferir valor a través del proceso de desarrollo de software.

## gemelo digital

Representación virtual de un sistema del mundo real, como un edificio, una fábrica, un equipo industrial o una línea de producción. Los gemelos digitales son compatibles con el mantenimiento predictivo, la supervisión remota y la optimización de la producción.

## tabla de dimensiones

En un [esquema en estrella](#), tabla más pequeña que contiene los atributos de datos sobre los datos cuantitativos de una tabla de hechos. Los atributos de la tabla de dimensiones suelen ser campos de texto o números discretos que se comportan como texto. Estos atributos se utilizan habitualmente para restringir consultas, filtrar y etiquetar conjuntos de resultados.

## desastre

Un evento que impide que una carga de trabajo o un sistema cumplan sus objetivos empresariales en su ubicación principal de implementación. Estos eventos pueden ser desastres naturales, fallos técnicos o el resultado de acciones humanas, como una configuración incorrecta involuntaria o un ataque de malware.

## recuperación de desastres (DR)

La estrategia y el proceso que se utilizan para minimizar el tiempo de inactividad y la pérdida de datos ocasionados por un [desastre](#). Para obtener más información, consulte [Recuperación ante desastres de cargas de trabajo en AWS: Recovery in the Cloud in the AWS Well-Architected Framework](#).

## DML

Consulte el lenguaje de manipulación de [bases de datos](#).

## diseño basado en el dominio

Un enfoque para desarrollar un sistema de software complejo mediante la conexión de sus componentes a dominios en evolución, o a los objetivos empresariales principales, a los que sirve cada componente. Este concepto lo introdujo Eric Evans en su libro, *Diseño impulsado por el dominio: abordando la complejidad en el corazón del software* (Boston: Addison-Wesley Professional, 2003). Para obtener información sobre cómo utilizar el diseño basado en dominios con el patrón de higos estranguladores, consulte [Modernización gradual de los servicios web antiguos de Microsoft ASP.NET \(ASMX\) mediante contenedores y Amazon API Gateway](#).

## DR

Consulte [recuperación ante desastres](#).

## detección de deriva

Seguimiento de las desviaciones con respecto a una configuración de referencia. Por ejemplo, puedes usarlo AWS CloudFormation para [detectar desviaciones en los recursos del sistema](#) o puedes usarlo AWS Control Tower para [detectar cambios en tu landing zone](#) que puedan afectar al cumplimiento de los requisitos de gobierno.

## DVSM

Consulte [el mapeo del flujo de valor del desarrollo](#).

## E

### EDA

Consulte el [análisis exploratorio de datos](#).

### EDI

Véase [intercambio electrónico de datos](#).

## computación en la periferia

La tecnología que aumenta la potencia de cálculo de los dispositivos inteligentes en la periferia de una red de IoT. En comparación con [la computación en nube](#), [la computación](#) perimetral puede reducir la latencia de la comunicación y mejorar el tiempo de respuesta.

## intercambio electrónico de datos (EDI)

El intercambio automatizado de documentos comerciales entre organizaciones. Para obtener más información, consulte [Qué es el intercambio electrónico de datos](#).

## cifrado

Proceso informático que transforma datos de texto plano, legibles por humanos, en texto cifrado.

## clave de cifrado

Cadena criptográfica de bits aleatorios que se genera mediante un algoritmo de cifrado. Las claves pueden variar en longitud y cada una se ha diseñado para ser impredecible y única.

## endianidad

El orden en el que se almacenan los bytes en la memoria del ordenador. Los sistemas big-endianos almacenan primero el byte más significativo. Los sistemas Little-Endian almacenan primero el byte menos significativo.

## punto de conexión

[Consulte el punto final del servicio](#).

## servicio de punto de conexión

Servicio que puede alojar en una nube privada virtual (VPC) para compartir con otros usuarios. Puede crear un servicio de punto final AWS PrivateLink y conceder permisos a otros directores Cuentas de AWS o a AWS Identity and Access Management (IAM). Estas cuentas o entidades principales pueden conectarse a su servicio de punto de conexión de forma privada mediante la creación de puntos de conexión de VPC de interfaz. Para obtener más información, consulte [Creación de un servicio de punto de conexión](#) en la documentación de Amazon Virtual Private Cloud (Amazon VPC).

## planificación de recursos empresariales (ERP)

Un sistema que automatiza y gestiona los procesos empresariales clave (como la contabilidad, el [MES](#) y la gestión de proyectos) de una empresa.

## cifrado de sobre

El proceso de cifrar una clave de cifrado con otra clave de cifrado. Para obtener más información, consulte el [cifrado de sobres](#) en la documentación de AWS Key Management Service (AWS KMS).

## entorno

Una instancia de una aplicación en ejecución. Los siguientes son los tipos de entornos más comunes en la computación en la nube:

- entorno de desarrollo: instancia de una aplicación en ejecución que solo se encuentra disponible para el equipo principal responsable del mantenimiento de la aplicación. Los entornos de desarrollo se utilizan para probar los cambios antes de promocionarlos a los entornos superiores. Este tipo de entorno a veces se denomina entorno de prueba.
- entornos inferiores: todos los entornos de desarrollo de una aplicación, como los que se utilizan para las compilaciones y pruebas iniciales.
- entorno de producción: instancia de una aplicación en ejecución a la que pueden acceder los usuarios finales. En un CI/CD proceso, el entorno de producción es el último entorno de implementación.
- entornos superiores: todos los entornos a los que pueden acceder usuarios que no sean del equipo de desarrollo principal. Esto puede incluir un entorno de producción, entornos de preproducción y entornos para las pruebas de aceptación por parte de los usuarios.

## epopeya

En las metodologías ágiles, son categorías funcionales que ayudan a organizar y priorizar el trabajo. Las epopeyas brindan una descripción detallada de los requisitos y las tareas de implementación. Por ejemplo, las epopeyas AWS de seguridad de CAF incluyen la gestión de identidades y accesos, los controles de detección, la seguridad de la infraestructura, la protección de datos y la respuesta a incidentes. Para obtener más información sobre las epopeyas en la estrategia de migración de AWS , consulte la [Guía de implementación del programa](#).

## PERP

Consulte [planificación de recursos empresariales](#).

## análisis de datos de tipo exploratorio (EDA)

El proceso de analizar un conjunto de datos para comprender sus características principales. Se recopilan o agregan datos y, a continuación, se realizan las investigaciones iniciales para

encontrar patrones, detectar anomalías y comprobar las suposiciones. El EDA se realiza mediante el cálculo de estadísticas resumidas y la creación de visualizaciones de datos.

## F

### tabla de datos

La tabla central de un [esquema en forma de estrella](#). Almacena datos cuantitativos sobre las operaciones comerciales. Normalmente, una tabla de hechos contiene dos tipos de columnas: las que contienen medidas y las que contienen una clave externa para una tabla de dimensiones.

### fallan rápidamente

Una filosofía que utiliza pruebas frecuentes e incrementales para reducir el ciclo de vida del desarrollo. Es una parte fundamental de un enfoque ágil.

### límite de aislamiento de fallas

En el Nube de AWS, un límite, como una zona de disponibilidad Región de AWS, un plano de control o un plano de datos, que limita el efecto de una falla y ayuda a mejorar la resiliencia de las cargas de trabajo. Para obtener más información, consulte [Límites de AWS aislamiento de errores](#).

### rama de característica

Consulte la [sucursal](#).

### características

Los datos de entrada que se utilizan para hacer una predicción. Por ejemplo, en un contexto de fabricación, las características pueden ser imágenes que se capturan periódicamente desde la línea de fabricación.

### importancia de las características

La importancia que tiene una característica para las predicciones de un modelo. Por lo general, esto se expresa como una puntuación numérica que se puede calcular mediante diversas técnicas, como las explicaciones aditivas de Shapley (SHAP) y los gradientes integrados. Para obtener más información, consulte [Interpretabilidad del modelo de aprendizaje automático con AWS](#).

## transformación de funciones

Optimizar los datos para el proceso de ML, lo que incluye enriquecer los datos con fuentes adicionales, escalar los valores o extraer varios conjuntos de información de un solo campo de datos. Esto permite que el modelo de ML se beneficie de los datos. Por ejemplo, si divide la fecha del “27 de mayo de 2021 00:15:37” en “jueves”, “mayo”, “2021” y “15”, puede ayudar al algoritmo de aprendizaje a aprender patrones matizados asociados a los diferentes componentes de los datos.

### indicaciones de pocos pasos

Proporcionar a un [LLM](#) un pequeño número de ejemplos que demuestren la tarea y el resultado deseado antes de pedirle que realice una tarea similar. Esta técnica es una aplicación del aprendizaje contextual, en el que los modelos aprenden a partir de ejemplos (planos) integrados en las instrucciones. Las indicaciones con pocas tomas pueden ser eficaces para tareas que requieren un formato, un razonamiento o un conocimiento del dominio específicos. [Consulte también el apartado de mensajes sin intervención.](#)

## FGAC

Consulte el control [de acceso detallado](#).

### control de acceso preciso (FGAC)

El uso de varias condiciones que tienen por objetivo permitir o denegar una solicitud de acceso.

### migración relámpago

Método de migración de bases de datos que utiliza la replicación continua de datos mediante la [captura de datos modificados](#) para migrar los datos en el menor tiempo posible, en lugar de utilizar un enfoque gradual. El objetivo es reducir al mínimo el tiempo de inactividad.

## FM

Consulte el [modelo básico](#).

### modelo de base (FM)

Una gran red neuronal de aprendizaje profundo que se ha estado entrenando con conjuntos de datos masivos de datos generalizados y sin etiquetar. FMs son capaces de realizar una amplia variedad de tareas generales, como comprender el lenguaje, generar texto e imágenes y conversar en lenguaje natural. Para obtener más información, consulte [Qué son los modelos básicos](#).

# G

## IA generativa

Un subconjunto de modelos de [IA](#) que se han entrenado con grandes cantidades de datos y que pueden utilizar un simple mensaje de texto para crear contenido y artefactos nuevos, como imágenes, vídeos, texto y audio. Para obtener más información, consulte [Qué es la IA generativa](#).

## bloqueo geográfico

Consulta [las restricciones geográficas](#).

## restricciones geográficas (bloqueo geográfico)

En Amazon CloudFront, una opción para impedir que los usuarios de países específicos accedan a las distribuciones de contenido. Puede utilizar una lista de permitidos o bloqueados para especificar los países aprobados y prohibidos. Para obtener más información, consulta [la sección Restringir la distribución geográfica del contenido](#) en la CloudFront documentación.

## Flujo de trabajo de Gitflow

Un enfoque en el que los entornos inferiores y superiores utilizan diferentes ramas en un repositorio de código fuente. El flujo de trabajo de Gitflow se considera heredado, y el [flujo de trabajo basado en enlaces troncales](#) es el enfoque moderno preferido.

## imagen dorada

Instantánea de un sistema o software que se utiliza como plantilla para implementar nuevas instancias de ese sistema o software. Por ejemplo, en la fabricación, una imagen dorada se puede utilizar para aprovisionar software en varios dispositivos y ayuda a mejorar la velocidad, la escalabilidad y la productividad de las operaciones de fabricación de dispositivos.

## estrategia de implementación desde cero

La ausencia de infraestructura existente en un entorno nuevo. Al adoptar una estrategia de implementación desde cero para una arquitectura de sistemas, puede seleccionar todas las tecnologías nuevas sin que estas deban ser compatibles con una infraestructura existente, lo que también se conoce como [implementación sobre infraestructura existente](#). Si está ampliando la infraestructura existente, puede combinar las estrategias de implementación sobre infraestructuras existentes y de implementación desde cero.

## barrera de protección

Una regla de alto nivel que ayuda a regular los recursos, las políticas y el cumplimiento en todas las unidades organizativas (OUs). Las barreras de protección preventivas aplican políticas para garantizar la alineación con los estándares de conformidad. Se implementan mediante políticas de control de servicios y límites de permisos de IAM. Las barreras de protección de detección detectan las vulneraciones de las políticas y los problemas de conformidad, y generan alertas para su corrección. Se implementan mediante Amazon AWS Config AWS Security Hub GuardDuty AWS Trusted Advisor, Amazon Inspector y AWS Lambda cheques personalizados.

# H

## HA

Consulte la [alta disponibilidad](#).

## migración heterogénea de bases de datos

Migración de la base de datos de origen a una base de datos de destino que utilice un motor de base de datos diferente (por ejemplo, de Oracle a Amazon Aurora). La migración heterogénea suele ser parte de un esfuerzo de rediseño de la arquitectura y convertir el esquema puede ser una tarea compleja. [AWS ofrece AWS SCT](#), lo cual ayuda con las conversiones de esquemas.

## alta disponibilidad (HA)

La capacidad de una carga de trabajo para funcionar de forma continua, sin intervención, en caso de desafíos o desastres. Los sistemas de alta disponibilidad están diseñados para realizar una conmutación por error automática, ofrecer un rendimiento de alta calidad de forma constante y gestionar diferentes cargas y fallos con un impacto mínimo en el rendimiento.

## modernización histórica

Un enfoque utilizado para modernizar y actualizar los sistemas de tecnología operativa (TO) a fin de satisfacer mejor las necesidades de la industria manufacturera. Un histórico es un tipo de base de datos que se utiliza para recopilar y almacenar datos de diversas fuentes en una fábrica.

## datos retenidos

Parte de los datos históricos etiquetados que se ocultan de un conjunto de datos que se utiliza para entrenar un modelo de aprendizaje [automático](#). Puede utilizar los datos de reserva para evaluar el rendimiento del modelo comparando las predicciones del modelo con los datos de reserva.

## migración homogénea de bases de datos

Migración de la base de datos de origen a una base de datos de destino que comparte el mismo motor de base de datos (por ejemplo, Microsoft SQL Server a Amazon RDS para SQL Server). La migración homogénea suele formar parte de un esfuerzo para volver a alojar o redefinir la plataforma. Puede utilizar las utilidades de bases de datos nativas para migrar el esquema.

## datos recientes

Datos a los que se accede con frecuencia, como datos en tiempo real o datos traslacionales recientes. Por lo general, estos datos requieren un nivel o una clase de almacenamiento de alto rendimiento para proporcionar respuestas rápidas a las consultas.

## hotfix

Una solución urgente para un problema crítico en un entorno de producción. Debido a su urgencia, las revisiones suelen realizarse fuera del flujo de trabajo habitual de las versiones.

## DevOps

## periodo de hiperatención

Periodo, inmediatamente después de la transición, durante el cual un equipo de migración administra y monitorea las aplicaciones migradas en la nube para solucionar cualquier problema. Por lo general, este periodo dura de 1 a 4 días. Al final del periodo de hiperatención, el equipo de migración suele transferir la responsabilidad de las aplicaciones al equipo de operaciones en la nube.

## I

## laC

Vea [la infraestructura como código](#).

## políticas basadas en identidades

Política asociada a uno o más directores de IAM que define sus permisos en el Nube de AWS entorno.

## aplicación inactiva

Aplicación que utiliza un promedio de CPU y memoria de entre 5 y 20 por ciento durante un periodo de 90 días. En un proyecto de migración, es habitual retirar estas aplicaciones o mantenerlas en las instalaciones.

## IloT

Consulte [Internet de las cosas industrial](#).

### infraestructura inmutable

Un modelo que implementa una nueva infraestructura para las cargas de trabajo de producción en lugar de actualizar, parchear o modificar la infraestructura existente. [Las infraestructuras inmutables son intrínsecamente más consistentes, fiables y predecibles que las infraestructuras mutables](#). Para obtener más información, consulte las prácticas recomendadas para [implementar con una infraestructura inmutable](#) en Well-Architected Framework AWS .

### VPC entrante (de entrada)

En una arquitectura de AWS cuentas múltiples, una VPC que acepta, inspecciona y enruta las conexiones de red desde fuera de una aplicación. La [arquitectura AWS de referencia de seguridad](#) recomienda configurar la cuenta de red con entradas, salidas e inspección VPCs para proteger la interfaz bidireccional entre la aplicación y el resto de Internet.

### migración gradual

Estrategia de transición en la que se migra la aplicación en partes pequeñas en lugar de realizar una transición única y completa. Por ejemplo, puede trasladar inicialmente solo unos pocos microservicios o usuarios al nuevo sistema. Tras comprobar que todo funciona correctamente, puede trasladar microservicios o usuarios adicionales de forma gradual hasta que pueda retirar su sistema heredado. Esta estrategia reduce los riesgos asociados a las grandes migraciones.

### Industria 4.0

Un término que [Klaus Schwab](#) introdujo en 2016 para referirse a la modernización de los procesos de fabricación mediante avances en la conectividad, los datos en tiempo real, la automatización, el análisis y la inteligencia artificial/aprendizaje automático.

### infraestructura

Todos los recursos y activos que se encuentran en el entorno de una aplicación.

### infraestructura como código (IaC)

Proceso de aprovisionamiento y administración de la infraestructura de una aplicación mediante un conjunto de archivos de configuración. La IaC se ha diseñado para ayudarlo a centralizar la administración de la infraestructura, estandarizar los recursos y escalar con rapidez a fin de que los entornos nuevos sean repetibles, fiables y consistentes.

## Internet de las cosas industrial (T) Ilo

El uso de sensores y dispositivos conectados a Internet en los sectores industriales, como el productivo, el eléctrico, el automotriz, el sanitario, el de las ciencias de la vida y el de la agricultura. Para obtener más información, consulte [Creación de una estrategia de transformación digital de la Internet de las cosas \(IIoT\) industrial](#).

## VPC de inspección

En una arquitectura de AWS cuentas múltiples, una VPC centralizada que gestiona las inspecciones del tráfico de red VPCs entre Internet y las redes locales (en una misma o Regiones de AWS diferente). La [arquitectura AWS de referencia de seguridad](#) recomienda configurar su cuenta de red con entrada, salida e inspección VPCs para proteger la interfaz bidireccional entre la aplicación e Internet en general.

## Internet de las cosas (IoT)

Red de objetos físicos conectados con sensores o procesadores integrados que se comunican con otros dispositivos y sistemas a través de Internet o de una red de comunicación local. Para obtener más información, consulte [¿Qué es IoT?](#).

## interpretabilidad

Característica de un modelo de machine learning que describe el grado en que un ser humano puede entender cómo las predicciones del modelo dependen de sus entradas. Para obtener más información, consulte Interpretabilidad del [modelo de aprendizaje automático](#) con AWS

## IoT

Consulte [Internet de las cosas](#).

## biblioteca de información de TI (ITIL)

Conjunto de prácticas recomendadas para ofrecer servicios de TI y alinearlos con los requisitos empresariales. La ITIL proporciona la base para la ITSM.

## administración de servicios de TI (ITSM)

Actividades asociadas con el diseño, la implementación, la administración y el soporte de los servicios de TI para una organización. Para obtener información sobre la integración de las operaciones en la nube con las herramientas de ITSM, consulte la [Guía de integración de operaciones](#).

## ITIL

Consulte la [biblioteca de información de TI](#).

## ITSM

Consulte [Administración de servicios de TI](#).

## L

### control de acceso basado en etiquetas (LBAC)

Una implementación del control de acceso obligatorio (MAC) en la que a los usuarios y a los propios datos se les asigna explícitamente un valor de etiqueta de seguridad. La intersección entre la etiqueta de seguridad del usuario y la etiqueta de seguridad de los datos determina qué filas y columnas puede ver el usuario.

### zona de aterrizaje

Una landing zone es un AWS entorno multicuenta bien diseñado, escalable y seguro. Este es un punto de partida desde el cual las empresas pueden lanzar e implementar rápidamente cargas de trabajo y aplicaciones con confianza en su entorno de seguridad e infraestructura. Para obtener más información sobre las zonas de aterrizaje, consulte [Configuración de un entorno de AWS seguro y escalable con varias cuentas](#).

### modelo de lenguaje grande (LLM)

Un modelo de [IA](#) de aprendizaje profundo que se entrena previamente con una gran cantidad de datos. Un LLM puede realizar múltiples tareas, como responder preguntas, resumir documentos, traducir textos a otros idiomas y completar oraciones. [Para obtener más información, consulte Qué son. LLMs](#)

### migración grande

Migración de 300 servidores o más.

### LBAC

Consulte control de [acceso basado en etiquetas](#).

### privilegio mínimo

La práctica recomendada de seguridad que consiste en conceder los permisos mínimos necesarios para realizar una tarea. Para obtener más información, consulte [Aplicar permisos de privilegio mínimo](#) en la documentación de IAM.

### migrar mediante lift-and-shift

Ver [7 Rs](#).

## sistema little-endian

Un sistema que almacena primero el byte menos significativo. Véase también [endianness](#).

## LLM

Véase un modelo de lenguaje [amplio](#).

## entornos inferiores

Véase [entorno](#).

# M

## machine learning (ML)

Un tipo de inteligencia artificial que utiliza algoritmos y técnicas para el reconocimiento y el aprendizaje de patrones. El ML analiza y aprende de los datos registrados, como los datos del Internet de las cosas (IoT), para generar un modelo estadístico basado en patrones. Para más información, consulte [Machine learning](#).

## rama principal

Ver [sucursal](#).

## malware

Software diseñado para comprometer la seguridad o la privacidad de la computadora. El malware puede interrumpir los sistemas informáticos, filtrar información confidencial u obtener acceso no autorizado. Algunos ejemplos de malware son los virus, los gusanos, el ransomware, los troyanos, el spyware y los keyloggers.

## servicios gestionados

Servicios de AWS para los que AWS opera la capa de infraestructura, el sistema operativo y las plataformas, y usted accede a los puntos finales para almacenar y recuperar datos. Amazon Simple Storage Service (Amazon S3) y Amazon DynamoDB son ejemplos de servicios gestionados. También se conocen como servicios abstractos.

## sistema de ejecución de fabricación (MES)

Un sistema de software para rastrear, monitorear, documentar y controlar los procesos de producción que convierten las materias primas en productos terminados en el taller.

## MAP

Consulte [Migration Acceleration Program](#).

### mecanismo

Un proceso completo en el que se crea una herramienta, se impulsa su adopción y, a continuación, se inspeccionan los resultados para realizar ajustes. Un mecanismo es un ciclo que se refuerza y mejora a sí mismo a medida que funciona. Para obtener más información, consulte [Creación de mecanismos](#) en el AWS Well-Architected Framework.

### cuenta de miembro

Todas las Cuentas de AWS demás cuentas, excepto la de administración, que forman parte de una organización. AWS Organizations Una cuenta no puede pertenecer a más de una organización a la vez.

## MES

Consulte el [sistema de ejecución de la fabricación](#).

### Transporte telemétrico de Message Queue Queue (MQTT)

[Un protocolo de comunicación ligero machine-to-machine \(M2M\), basado en el patrón de publicación/suscripción, para dispositivos de IoT con recursos limitados.](#)

### microservicio

Un servicio pequeño e independiente que se comunica a través de una red bien definida APIs y que, por lo general, es propiedad de equipos pequeños e independientes. Por ejemplo, un sistema de seguros puede incluir microservicios que se adapten a las capacidades empresariales, como las de ventas o marketing, o a subdominios, como las de compras, reclamaciones o análisis. Los beneficios de los microservicios incluyen la agilidad, la escalabilidad flexible, la facilidad de implementación, el código reutilizable y la resiliencia. Para obtener más información, consulte [Integrar microservicios mediante AWS servicios sin servidor](#).

### arquitectura de microservicios

Un enfoque para crear una aplicación con componentes independientes que ejecutan cada proceso de la aplicación como un microservicio. Estos microservicios se comunican a través de una interfaz bien definida mediante un uso ligero. APIs Cada microservicio de esta arquitectura se puede actualizar, implementar y escalar para satisfacer la demanda de funciones específicas de una aplicación. Para obtener más información, consulte [Implementación de microservicios](#) en AWS

## Programa de aceleración de la migración (MAP)

Un AWS programa que proporciona soporte de consultoría, formación y servicios para ayudar a las organizaciones a crear una base operativa sólida para migrar a la nube y para ayudar a compensar el costo inicial de las migraciones. El MAP incluye una metodología de migración para ejecutar las migraciones antiguas de forma metódica y un conjunto de herramientas para automatizar y acelerar los escenarios de migración más comunes.

### migración a escala

Proceso de transferencia de la mayoría de la cartera de aplicaciones a la nube en oleadas, con más aplicaciones desplazadas a un ritmo más rápido en cada oleada. En esta fase, se utilizan las prácticas recomendadas y las lecciones aprendidas en las fases anteriores para implementar una fábrica de migración de equipos, herramientas y procesos con el fin de agilizar la migración de las cargas de trabajo mediante la automatización y la entrega ágil. Esta es la tercera fase de la [estrategia de migración de AWS](#).

### fábrica de migración

Equipos multifuncionales que agilizan la migración de las cargas de trabajo mediante enfoques automatizados y ágiles. Los equipos de las fábricas de migración suelen incluir a analistas y propietarios de operaciones, empresas, ingenieros de migración, desarrolladores y DevOps profesionales que trabajan a pasos agigantados. Entre el 20 y el 50 por ciento de la cartera de aplicaciones empresariales se compone de patrones repetidos que pueden optimizarse mediante un enfoque de fábrica. Para obtener más información, consulte la [discusión sobre las fábricas de migración](#) y la [Guía de fábricas de migración a la nube](#) en este contenido.

### metadatos de migración

Información sobre la aplicación y el servidor que se necesita para completar la migración. Cada patrón de migración requiere un conjunto diferente de metadatos de migración. Algunos ejemplos de metadatos de migración son la subred de destino, el grupo de seguridad y AWS la cuenta.

### patrón de migración

Tarea de migración repetible que detalla la estrategia de migración, el destino de la migración y la aplicación o el servicio de migración utilizados. Ejemplo: realoje la migración a Amazon EC2 con AWS Application Migration Service.

## Migration Portfolio Assessment (MPA)

Una herramienta en línea que proporciona información para validar el modelo de negocio para migrar a. Nube de AWS La MPA ofrece una evaluación detallada de la cartera (adecuación del

tamaño de los servidores, precios, comparaciones del costo total de propiedad, análisis de los costos de migración), así como una planificación de la migración (análisis y recopilación de datos de aplicaciones, agrupación de aplicaciones, priorización de la migración y planificación de oleadas). La [herramienta MPA](#) (requiere iniciar sesión) está disponible de forma gratuita para todos los AWS consultores y consultores asociados de APN.

### Evaluación de la preparación para la migración (MRA)

Proceso que consiste en obtener información sobre el estado de preparación de una organización para la nube, identificar sus puntos fuertes y débiles y elaborar un plan de acción para cerrar las brechas identificadas mediante el AWS CAF. Para obtener más información, consulte la [Guía de preparación para la migración](#). La MRA es la primera fase de la [estrategia de migración de AWS](#).

### estrategia de migración

El enfoque utilizado para migrar una carga de trabajo a. Nube de AWS Para obtener más información, consulte la entrada de las [7 R](#) de este glosario y consulte [Movilice a su organización para acelerar las migraciones a gran escala](#).

### ML

[Consulte el aprendizaje automático.](#)

### modernización

Transformar una aplicación obsoleta (antigua o monolítica) y su infraestructura en un sistema ágil, elástico y de alta disponibilidad en la nube para reducir los gastos, aumentar la eficiencia y aprovechar las innovaciones. Para obtener más información, consulte [Estrategia para modernizar las aplicaciones en el Nube de AWS](#).

### evaluación de la preparación para la modernización

Evaluación que ayuda a determinar la preparación para la modernización de las aplicaciones de una organización; identifica los beneficios, los riesgos y las dependencias; y determina qué tan bien la organización puede soportar el estado futuro de esas aplicaciones. El resultado de la evaluación es un esquema de la arquitectura objetivo, una hoja de ruta que detalla las fases de desarrollo y los hitos del proceso de modernización y un plan de acción para abordar las brechas identificadas. Para obtener más información, consulte [Evaluación de la preparación para la modernización de las aplicaciones en el Nube de AWS](#).

### aplicaciones monolíticas (monolitos)

Aplicaciones que se ejecutan como un único servicio con procesos estrechamente acoplados. Las aplicaciones monolíticas presentan varios inconvenientes. Si una característica de la

aplicación experimenta un aumento en la demanda, se debe escalar toda la arquitectura. Agregar o mejorar las características de una aplicación monolítica también se vuelve más complejo a medida que crece la base de código. Para solucionar problemas con la aplicación, puede utilizar una arquitectura de microservicios. Para obtener más información, consulte [Descomposición de monolitos en microservicios](#).

## MAPA

Consulte [la evaluación de la cartera de migración](#).

## MQTT

Consulte [Message Queue Queue Telemetría](#) y Transporte.

## clasificación multiclase

Un proceso que ayuda a generar predicciones para varias clases (predice uno de más de dos resultados). Por ejemplo, un modelo de ML podría preguntar “¿Este producto es un libro, un automóvil o un teléfono?” o “¿Qué categoría de productos es más interesante para este cliente?”.

## infraestructura mutable

Un modelo que actualiza y modifica la infraestructura existente para las cargas de trabajo de producción. Para mejorar la coherencia, la fiabilidad y la previsibilidad, el AWS Well-Architected Framework recomienda el uso [de una infraestructura inmutable](#) como práctica recomendada.

## O

### OAC

[Consulte el control de acceso de origen](#).

### OAI

Consulte la [identidad de acceso de origen](#).

### OCM

Consulte [gestión del cambio organizacional](#).

## migración fuera de línea

Método de migración en el que la carga de trabajo de origen se elimina durante el proceso de migración. Este método implica un tiempo de inactividad prolongado y, por lo general, se utiliza para cargas de trabajo pequeñas y no críticas.

## OI

Consulte [integración de operaciones](#).

## OLA

Véase el [acuerdo a nivel operativo](#).

## migración en línea

Método de migración en el que la carga de trabajo de origen se copia al sistema de destino sin que se desconecte. Las aplicaciones que están conectadas a la carga de trabajo pueden seguir funcionando durante la migración. Este método implica un tiempo de inactividad nulo o mínimo y, por lo general, se utiliza para cargas de trabajo de producción críticas.

## OPC-UA

Consulte [Open Process Communications: arquitectura unificada](#).

## Comunicaciones de proceso abierto: arquitectura unificada (OPC-UA)

Un protocolo de comunicación machine-to-machine (M2M) para la automatización industrial. El OPC-UA proporciona un estándar de interoperabilidad con esquemas de cifrado, autenticación y autorización de datos.

## acuerdo de nivel operativo (OLA)

Acuerdo que aclara lo que los grupos de TI operativos se comprometen a ofrecerse entre sí, para respaldar un acuerdo de nivel de servicio (SLA).

## revisión de la preparación operativa (ORR)

Una lista de preguntas y las mejores prácticas asociadas que le ayudan a comprender, evaluar, prevenir o reducir el alcance de los incidentes y posibles fallos. Para obtener más información, consulte las [Revisiones de preparación operativa \(ORR\)](#) en AWS Well-Architected Framework.

## tecnología operativa (OT)

Sistemas de hardware y software que funcionan con el entorno físico para controlar las operaciones, los equipos y la infraestructura industriales. En la industria manufacturera, la integración de los sistemas de TO y tecnología de la información (TI) es un enfoque clave para las transformaciones de [la industria 4.0](#).

## integración de operaciones (OI)

Proceso de modernización de las operaciones en la nube, que implica la planificación de la preparación, la automatización y la integración. Para obtener más información, consulte la [Guía de integración de las operaciones](#).

## registro de seguimiento organizativo

Un registro creado por el AWS CloudTrail que se registran todos los eventos para todos Cuentas de AWS los miembros de una organización AWS Organizations. Este registro de seguimiento se crea en cada Cuenta de AWS que forma parte de la organización y realiza un seguimiento de la actividad en cada cuenta. Para obtener más información, consulte [Crear un registro para una organización](#) en la CloudTrail documentación.

## administración del cambio organizacional (OCM)

Marco para administrar las transformaciones empresariales importantes y disruptivas desde la perspectiva de las personas, la cultura y el liderazgo. La OCM ayuda a las empresas a prepararse para nuevos sistemas y estrategias y a realizar la transición a ellos, al acelerar la adopción de cambios, abordar los problemas de transición e impulsar cambios culturales y organizacionales. En la estrategia de AWS migración, este marco se denomina aceleración de personal, debido a la velocidad de cambio que requieren los proyectos de adopción de la nube. Para obtener más información, consulte la [Guía de OCM](#).

## control de acceso de origen (OAC)

En CloudFront, una opción mejorada para restringir el acceso y proteger el contenido del Amazon Simple Storage Service (Amazon S3). El OAC admite todos los buckets de S3 Regiones de AWS, el cifrado del lado del servidor AWS KMS (SSE-KMS) y las solicitudes dinámicas PUT y DELETE dirigidas al bucket de S3.

## identidad de acceso de origen (OAI)

En CloudFront, una opción para restringir el acceso y proteger el contenido de Amazon S3. Cuando utiliza OAI, CloudFront crea un principal con el que Amazon S3 puede autenticarse. Los directores autenticados solo pueden acceder al contenido de un bucket de S3 a través de una distribución específica. CloudFront Consulte también el [OAC](#), que proporciona un control de acceso más detallado y mejorado.

## ORR

Consulte la revisión de [la preparación operativa](#).

## OT

Consulte la [tecnología operativa](#).

## VPC saliente (de salida)

En una arquitectura de AWS cuentas múltiples, una VPC que gestiona las conexiones de red que se inician desde una aplicación. La [arquitectura AWS de referencia de seguridad](#) recomienda configurar la cuenta de red con entradas, salidas e inspección VPCs para proteger la interfaz bidireccional entre la aplicación e Internet en general.

## P

### límite de permisos

Una política de administración de IAM que se adjunta a las entidades principales de IAM para establecer los permisos máximos que puede tener el usuario o el rol. Para obtener más información, consulte [Límites de permisos](#) en la documentación de IAM.

### información de identificación personal (PII)

Información que, vista directamente o combinada con otros datos relacionados, puede utilizarse para deducir de manera razonable la identidad de una persona. Algunos ejemplos de información de identificación personal son los nombres, las direcciones y la información de contacto.

## PII

Consulte la [información de identificación personal](#).

### manual de estrategias

Conjunto de pasos predefinidos que capturan el trabajo asociado a las migraciones, como la entrega de las funciones de operaciones principales en la nube. Un manual puede adoptar la forma de scripts, manuales de procedimientos automatizados o resúmenes de los procesos o pasos necesarios para operar un entorno modernizado.

## PLC

Consulte [controlador lógico programable](#).

## PLM

Consulte la [gestión del ciclo de vida del producto](#).

## policy

Un objeto que puede definir los permisos (consulte la [política basada en la identidad](#)), especifique las condiciones de acceso (consulte la [política basada en los recursos](#)) o defina los permisos máximos para todas las cuentas de una organización AWS Organizations (consulte la política de control de [servicios](#)).

## persistencia políglota

Elegir de forma independiente la tecnología de almacenamiento de datos de un microservicio en función de los patrones de acceso a los datos y otros requisitos. Si sus microservicios tienen la misma tecnología de almacenamiento de datos, pueden enfrentarse a desafíos de implementación o experimentar un rendimiento deficiente. Los microservicios se implementan más fácilmente y logran un mejor rendimiento y escalabilidad si utilizan el almacén de datos que mejor se adapte a sus necesidades. Para obtener más información, consulte [Habilitación de la persistencia de datos en los microservicios](#).

## evaluación de cartera

Proceso de detección, análisis y priorización de la cartera de aplicaciones para planificar la migración. Para obtener más información, consulte la [Evaluación de la preparación para la migración](#).

## predicate

Una condición de consulta que devuelve `true` o `false`, normalmente, se encuentra en una cláusula. `WHERE`

## pulsar un predicado

Técnica de optimización de consultas de bases de datos que filtra los datos de la consulta antes de transferirlos. Esto reduce la cantidad de datos que se deben recuperar y procesar de la base de datos relacional y mejora el rendimiento de las consultas.

## control preventivo

Un control de seguridad diseñado para evitar que ocurra un evento. Estos controles son la primera línea de defensa para evitar el acceso no autorizado o los cambios no deseados en la red. Para obtener más información, consulte [Controles preventivos](#) en Implementación de controles de seguridad en AWS.

## entidad principal

Una entidad AWS que puede realizar acciones y acceder a los recursos. Esta entidad suele ser un usuario raíz para un Cuenta de AWS rol de IAM o un usuario. Para obtener más información, consulte Entidad principal en [Términos y conceptos de roles](#) en la documentación de IAM.

## privacidad desde el diseño

Un enfoque de ingeniería de sistemas que tiene en cuenta la privacidad durante todo el proceso de desarrollo.

## zonas alojadas privadas

Un contenedor que contiene información sobre cómo desea que Amazon Route 53 responda a las consultas de DNS de un dominio y sus subdominios dentro de uno o más VPCs. Para obtener más información, consulte [Uso de zonas alojadas privadas](#) en la documentación de Route 53.

## control proactivo

Un [control de seguridad](#) diseñado para evitar el despliegue de recursos no conformes. Estos controles escanean los recursos antes de aprovisionarlos. Si el recurso no cumple con el control, significa que no está aprovisionado. Para obtener más información, consulte la [guía de referencia de controles](#) en la AWS Control Tower documentación y consulte [Controles proactivos](#) en Implementación de controles de seguridad en AWS.

## gestión del ciclo de vida del producto (PLM)

La gestión de los datos y los procesos de un producto a lo largo de todo su ciclo de vida, desde el diseño, el desarrollo y el lanzamiento, pasando por el crecimiento y la madurez, hasta el rechazo y la retirada.

## entorno de producción

Consulte [el entorno](#).

## controlador lógico programable (PLC)

En la fabricación, una computadora adaptable y altamente confiable que monitorea las máquinas y automatiza los procesos de fabricación.

## encadenamiento rápido

Utilizar la salida de una solicitud de [LLM](#) como entrada para la siguiente solicitud para generar mejores respuestas. Esta técnica se utiliza para dividir una tarea compleja en subtareas o para

refinar o ampliar de forma iterativa una respuesta preliminar. Ayuda a mejorar la precisión y la relevancia de las respuestas de un modelo y permite obtener resultados más detallados y personalizados.

## seudonimización

El proceso de reemplazar los identificadores personales de un conjunto de datos por valores de marcadores de posición. La seudonimización puede ayudar a proteger la privacidad personal. Los datos seudonimizados siguen considerándose datos personales.

## publish/subscribe (pub/sub)

Un patrón que permite las comunicaciones asíncronas entre microservicios para mejorar la escalabilidad y la capacidad de respuesta. Por ejemplo, en un [MES](#) basado en microservicios, un microservicio puede publicar mensajes de eventos en un canal al que se puedan suscribir otros microservicios. El sistema puede añadir nuevos microservicios sin cambiar el servicio de publicación.

## Q

### plan de consulta

Serie de pasos, como instrucciones, que se utilizan para acceder a los datos de un sistema de base de datos relacional SQL.

### regresión del plan de consulta

El optimizador de servicios de la base de datos elige un plan menos óptimo que antes de un cambio determinado en el entorno de la base de datos. Los cambios en estadísticas, restricciones, configuración del entorno, enlaces de parámetros de consultas y actualizaciones del motor de base de datos PostgreSQL pueden provocar una regresión del plan.

## R

### Matriz RACI

Véase [responsable, responsable, consultado, informado \(RACI\)](#).

### RAG

Consulte [Recuperación y generación aumentada](#).

## ransomware

Software malicioso que se ha diseñado para bloquear el acceso a un sistema informático o a los datos hasta que se efectúe un pago.

## Matriz RASCI

Véase [responsable, responsable, consultado, informado \(RACI\)](#).

## RCAC

Consulte control de [acceso por filas y columnas](#).

## réplica de lectura

Una copia de una base de datos que se utiliza con fines de solo lectura. Puede enrutar las consultas a la réplica de lectura para reducir la carga en la base de datos principal.

## rediseñar

Ver [7 Rs](#).

## objetivo de punto de recuperación (RPO)

La cantidad de tiempo máximo aceptable desde el último punto de recuperación de datos. Esto determina qué se considera una pérdida de datos aceptable entre el último punto de recuperación y la interrupción del servicio.

## objetivo de tiempo de recuperación (RTO)

La demora máxima aceptable entre la interrupción del servicio y el restablecimiento del servicio.

## refactorizar

Ver [7 Rs](#).

## Región

Una colección de AWS recursos en un área geográfica. Cada uno Región de AWS está aislado e independiente de los demás para proporcionar tolerancia a las fallas, estabilidad y resiliencia. Para obtener más información, consulte [Regiones de AWS Especificar qué cuenta puede usar](#).

## regresión

Una técnica de ML que predice un valor numérico. Por ejemplo, para resolver el problema de “¿A qué precio se venderá esta casa?”, un modelo de ML podría utilizar un modelo de regresión lineal para predecir el precio de venta de una vivienda en función de datos conocidos sobre ella (por ejemplo, los metros cuadrados).

volver a alojar

Consulte [7 Rs.](#)

versión

En un proceso de implementación, el acto de promover cambios en un entorno de producción. trasladarse

Ver [7 Rs.](#)

redefinir la plataforma

Ver [7 Rs.](#)

recompra

Ver [7 Rs.](#)

resiliencia

La capacidad de una aplicación para resistir las interrupciones o recuperarse de ellas. [La alta disponibilidad](#) y la [recuperación ante desastres](#) son consideraciones comunes a la hora de planificar la resiliencia en el. Nube de AWS Para obtener más información, consulte [Nube de AWS Resiliencia](#).

política basada en recursos

Una política asociada a un recurso, como un bucket de Amazon S3, un punto de conexión o una clave de cifrado. Este tipo de política especifica a qué entidades principales se les permite el acceso, las acciones compatibles y cualquier otra condición que deba cumplirse.

matriz responsable, confiable, consultada e informada (RACI)

Una matriz que define las funciones y responsabilidades de todas las partes involucradas en las actividades de migración y las operaciones de la nube. El nombre de la matriz se deriva de los tipos de responsabilidad definidos en la matriz: responsable (R), contable (A), consultado (C) e informado (I). El tipo de soporte (S) es opcional. Si incluye el soporte, la matriz se denomina matriz RASCI y, si la excluye, se denomina matriz RACI.

control receptivo

Un control de seguridad que se ha diseñado para corregir los eventos adversos o las desviaciones con respecto a su base de seguridad. Para obtener más información, consulte [Controles receptivos](#) en Implementación de controles de seguridad en AWS.

## retain

Consulte [7 Rs](#).

## jubilarse

Ver [7 Rs](#).

## Generación aumentada de recuperación (RAG)

Tecnología de [inteligencia artificial generativa](#) en la que un máster [hace referencia](#) a una fuente de datos autorizada que se encuentra fuera de sus fuentes de datos de formación antes de generar una respuesta. Por ejemplo, un modelo RAG podría realizar una búsqueda semántica en la base de conocimientos o en los datos personalizados de una organización. Para obtener más información, consulte [Qué es](#) el RAG.

## rotación

Proceso de actualizar periódicamente un [secreto](#) para dificultar el acceso de un atacante a las credenciales.

## control de acceso por filas y columnas (RCAC)

El uso de expresiones SQL básicas y flexibles que tienen reglas de acceso definidas. El RCAC consta de permisos de fila y máscaras de columnas.

## RPO

Consulte el [objetivo del punto de recuperación](#).

## RTO

Consulte el [objetivo de tiempo de recuperación](#).

## manual de procedimientos

Conjunto de procedimientos manuales o automatizados necesarios para realizar una tarea específica. Por lo general, se diseñan para agilizar las operaciones o los procedimientos repetitivos con altas tasas de error.

# S

## SAML 2.0

Un estándar abierto que utilizan muchos proveedores de identidad (IdPs). Esta función permite el inicio de sesión único (SSO) federado, de modo que los usuarios pueden iniciar sesión AWS

Management Console o llamar a las operaciones de la AWS API sin tener que crear un usuario en IAM para todos los miembros de la organización. Para obtener más información sobre la federación basada en SAML 2.0, consulte [Acerca de la federación basada en SAML 2.0](#) en la documentación de IAM.

## SCADA

Consulte el [control de supervisión y la adquisición de datos](#).

## SCP

Consulte la [política de control de servicios](#).

## secreta

Información confidencial o restringida, como una contraseña o credenciales de usuario, que almacene de forma cifrada. AWS Secrets Manager Se compone del valor secreto y sus metadatos. El valor secreto puede ser binario, una sola cadena o varias cadenas. Para obtener más información, consulta [¿Qué hay en un secreto de Secrets Manager?](#) en la documentación de Secrets Manager.

## seguridad desde el diseño

Un enfoque de ingeniería de sistemas que tiene en cuenta la seguridad durante todo el proceso de desarrollo.

## control de seguridad

Barrera de protección técnica o administrativa que impide, detecta o reduce la capacidad de un agente de amenazas para aprovechar una vulnerabilidad de seguridad. Existen cuatro tipos principales de controles de seguridad: [preventivos, de detección](#), con [capacidad](#) de [respuesta](#) y [proactivos](#).

## refuerzo de la seguridad

Proceso de reducir la superficie expuesta a ataques para hacerla más resistente a los ataques. Esto puede incluir acciones, como la eliminación de los recursos que ya no se necesitan, la implementación de prácticas recomendadas de seguridad consistente en conceder privilegios mínimos o la desactivación de características innecesarias en los archivos de configuración.

## sistema de información sobre seguridad y administración de eventos (SIEM)

Herramientas y servicios que combinan sistemas de administración de información sobre seguridad (SIM) y de administración de eventos de seguridad (SEM). Un sistema de SIEM

recopila, monitorea y analiza los datos de servidores, redes, dispositivos y otras fuentes para detectar amenazas y brechas de seguridad y generar alertas.

#### automatización de la respuesta de seguridad

Una acción predefinida y programada que está diseñada para responder automáticamente a un evento de seguridad o remediarlo. Estas automatizaciones sirven como controles de seguridad [detectables](#) o [adaptables](#) que le ayudan a implementar las mejores prácticas AWS de seguridad. Algunos ejemplos de acciones de respuesta automatizadas incluyen la modificación de un grupo de seguridad de VPC, la aplicación de parches a una EC2 instancia de Amazon o la rotación de credenciales.

#### cifrado del servidor

Cifrado de los datos en su destino, por parte de quien Servicio de AWS los recibe.

#### política de control de servicio (SCP)

Política que proporciona un control centralizado de los permisos de todas las cuentas de una organización en AWS Organizations. SCPs defina barreras o establezca límites a las acciones que un administrador puede delegar en usuarios o roles. Puede utilizarlas SCPs como listas de permitidos o rechazados para especificar qué servicios o acciones están permitidos o prohibidos. Para obtener más información, consulte [las políticas de control de servicios](#) en la AWS Organizations documentación.

#### punto de enlace de servicio

La URL del punto de entrada de un Servicio de AWS. Para conectarse mediante programación a un servicio de destino, puede utilizar un punto de conexión. Para obtener más información, consulte [Puntos de conexión de Servicio de AWS](#) en Referencia general de AWS.

#### acuerdo de nivel de servicio (SLA)

Acuerdo que aclara lo que un equipo de TI se compromete a ofrecer a los clientes, como el tiempo de actividad y el rendimiento del servicio.

#### indicador de nivel de servicio (SLI)

Medición de un aspecto del rendimiento de un servicio, como la tasa de errores, la disponibilidad o el rendimiento.

#### objetivo de nivel de servicio (SLO)

[Una métrica objetivo que representa el estado de un servicio, medido mediante un indicador de nivel de servicio.](#)

## modelo de responsabilidad compartida

Un modelo que describe la responsabilidad que compartes con respecto a la seguridad y AWS el cumplimiento de la nube. AWS es responsable de la seguridad de la nube, mientras que usted es responsable de la seguridad en la nube. Para obtener más información, consulte el [Modelo de responsabilidad compartida](#).

## SIEM

Consulte [la información de seguridad y el sistema de gestión de eventos](#).

## punto único de fallo (SPOF)

Una falla en un único componente crítico de una aplicación que puede interrumpir el sistema.

## SLA

Consulte el acuerdo [de nivel de servicio](#).

## SLI

Consulte el indicador de [nivel de servicio](#).

## SLO

Consulte el objetivo de nivel de [servicio](#).

## split-and-seed modelo

Un patrón para escalar y acelerar los proyectos de modernización. A medida que se definen las nuevas funciones y los lanzamientos de los productos, el equipo principal se divide para crear nuevos equipos de productos. Esto ayuda a ampliar las capacidades y los servicios de su organización, mejora la productividad de los desarrolladores y apoya la innovación rápida. Para obtener más información, consulte [Enfoque gradual para modernizar las aplicaciones en](#). Nube de AWS

## SPOT

Consulte el [punto único de falla](#).

## esquema en forma de estrella

Estructura organizativa de una base de datos que utiliza una tabla de datos grande para almacenar datos transaccionales o medidos y una o más tablas dimensionales más pequeñas para almacenar los atributos de los datos. Esta estructura está diseñada para usarse en un [almacén de datos](#) o con fines de inteligencia empresarial.

## patrón de higo estrangulador

Un enfoque para modernizar los sistemas monolíticos mediante la reescritura y el reemplazo gradual de las funciones del sistema hasta que se pueda dismantelar el sistema heredado. Este patrón utiliza la analogía de una higuera que crece hasta convertirse en un árbol estable y, finalmente, se apodera y reemplaza a su host. El patrón fue [presentado por Martin Fowler](#) como una forma de gestionar el riesgo al reescribir sistemas monolíticos. Para ver un ejemplo con la aplicación de este patrón, consulte [Modernización gradual de los servicios web antiguos de Microsoft ASP.NET \(ASMX\) mediante contenedores y Amazon API Gateway](#).

## subred

Un intervalo de direcciones IP en la VPC. Una subred debe residir en una sola zona de disponibilidad.

## supervisión, control y adquisición de datos (SCADA)

En la industria manufacturera, un sistema que utiliza hardware y software para monitorear los activos físicos y las operaciones de producción.

## cifrado simétrico

Un algoritmo de cifrado que utiliza la misma clave para cifrar y descifrar los datos.

## pruebas sintéticas

Probar un sistema de manera que simule las interacciones de los usuarios para detectar posibles problemas o monitorear el rendimiento. Puede usar [Amazon CloudWatch Synthetics](#) para crear estas pruebas.

## indicador del sistema

Una técnica para proporcionar contexto, instrucciones o pautas a un [LLM](#) para dirigir su comportamiento. Las indicaciones del sistema ayudan a establecer el contexto y las reglas para las interacciones con los usuarios.

# T

## etiquetas

Pares clave-valor que actúan como metadatos para organizar los recursos. AWS Las etiquetas pueden ayudarle a administrar, identificar, organizar, buscar y filtrar recursos. Para obtener más información, consulte [Etiquetado de los recursos de AWS](#).

## variable de destino

El valor que intenta predecir en el ML supervisado. Esto también se conoce como variable de resultado. Por ejemplo, en un entorno de fabricación, la variable objetivo podría ser un defecto del producto.

## lista de tareas

Herramienta que se utiliza para hacer un seguimiento del progreso mediante un manual de procedimientos. La lista de tareas contiene una descripción general del manual de procedimientos y una lista de las tareas generales que deben completarse. Para cada tarea general, se incluye la cantidad estimada de tiempo necesario, el propietario y el progreso.

## entorno de prueba

[Consulte entorno.](#)

## entrenamiento

Proporcionar datos de los que pueda aprender su modelo de ML. Los datos de entrenamiento deben contener la respuesta correcta. El algoritmo de aprendizaje encuentra patrones en los datos de entrenamiento que asignan los atributos de los datos de entrada al destino (la respuesta que desea predecir). Genera un modelo de ML que captura estos patrones. Luego, el modelo de ML se puede utilizar para obtener predicciones sobre datos nuevos para los que no se conoce el destino.

## puerta de enlace de tránsito

Un centro de tránsito de red que puede usar para interconectar sus VPCs redes con las locales. Para obtener más información, consulte [Qué es una pasarela de tránsito](#) en la AWS Transit Gateway documentación.

## flujo de trabajo basado en enlaces troncales

Un enfoque en el que los desarrolladores crean y prueban características de forma local en una rama de característica y, a continuación, combinan esos cambios en la rama principal. Luego, la rama principal se adapta a los entornos de desarrollo, preproducción y producción, de forma secuencial.

## acceso de confianza

Otorgar permisos a un servicio que especifique para realizar tareas en su organización AWS Organizations y en sus cuentas en su nombre. El servicio de confianza crea un rol vinculado al servicio en cada cuenta, cuando ese rol es necesario, para realizar las tareas de administración

por usted. Para obtener más información, consulte [AWS Organizations Utilización con otros AWS servicios](#) en la AWS Organizations documentación.

## ajuste

Cambiar aspectos de su proceso de formación a fin de mejorar la precisión del modelo de ML. Por ejemplo, puede entrenar el modelo de ML al generar un conjunto de etiquetas, incorporar etiquetas y, luego, repetir estos pasos varias veces con diferentes ajustes para optimizar el modelo.

## equipo de dos pizzas

Un DevOps equipo pequeño al que puedes alimentar con dos pizzas. Un equipo formado por dos integrantes garantiza la mejor oportunidad posible de colaboración en el desarrollo de software.

# U

## incertidumbre

Un concepto que hace referencia a información imprecisa, incompleta o desconocida que puede socavar la fiabilidad de los modelos predictivos de ML. Hay dos tipos de incertidumbre: la incertidumbre epistémica se debe a datos limitados e incompletos, mientras que la incertidumbre aleatoria se debe al ruido y la aleatoriedad inherentes a los datos. Para más información, consulte la guía [Cuantificación de la incertidumbre en los sistemas de aprendizaje profundo](#).

## tareas indiferenciadas

También conocido como tareas arduas, es el trabajo que es necesario para crear y operar una aplicación, pero que no proporciona un valor directo al usuario final ni proporciona una ventaja competitiva. Algunos ejemplos de tareas indiferenciadas son la adquisición, el mantenimiento y la planificación de la capacidad.

## entornos superiores

Ver [entorno](#).

## V

### succión

Una operación de mantenimiento de bases de datos que implica limpiar después de las actualizaciones incrementales para recuperar espacio de almacenamiento y mejorar el rendimiento.

### control de versión

Procesos y herramientas que realizan un seguimiento de los cambios, como los cambios en el código fuente de un repositorio.

### Emparejamiento de VPC

Una conexión entre dos VPCs que le permite enrutar el tráfico mediante direcciones IP privadas. Para obtener más información, consulte [¿Qué es una interconexión de VPC?](#) en la documentación de Amazon VPC.

### vulnerabilidad

Defecto de software o hardware que pone en peligro la seguridad del sistema.

## W

### caché caliente

Un búfer caché que contiene datos actuales y relevantes a los que se accede con frecuencia. La instancia de base de datos puede leer desde la caché del búfer, lo que es más rápido que leer desde la memoria principal o el disco.

### datos templados

Datos a los que el acceso es infrecuente. Al consultar este tipo de datos, normalmente se aceptan consultas moderadamente lentas.

### función de ventana

Función SQL que realiza un cálculo en un grupo de filas que se relacionan de alguna manera con el registro actual. Las funciones de ventana son útiles para procesar tareas, como calcular una media móvil o acceder al valor de las filas en función de la posición relativa de la fila actual.

## carga de trabajo

Conjunto de recursos y código que ofrece valor comercial, como una aplicación orientada al cliente o un proceso de backend.

## flujo de trabajo

Grupos funcionales de un proyecto de migración que son responsables de un conjunto específico de tareas. Cada flujo de trabajo es independiente, pero respalda a los demás flujos de trabajo del proyecto. Por ejemplo, el flujo de trabajo de la cartera es responsable de priorizar las aplicaciones, planificar las oleadas y recopilar los metadatos de migración. El flujo de trabajo de la cartera entrega estos recursos al flujo de trabajo de migración, que luego migra los servidores y las aplicaciones.

## GUSANO

Mira, [escribe una vez, lee muchas](#).

## WQF

Consulte el [marco AWS de calificación de la carga](#) de trabajo.

## escribe una vez, lee muchas (WORM)

Un modelo de almacenamiento que escribe los datos una sola vez y evita que se eliminen o modifiquen. Los usuarios autorizados pueden leer los datos tantas veces como sea necesario, pero no pueden cambiarlos. Esta infraestructura de almacenamiento de datos se considera [inmutable](#).

## Z

### ataque de día cero

Un ataque, normalmente de malware, que aprovecha una vulnerabilidad de [día cero](#).

### vulnerabilidad de día cero

Un defecto o una vulnerabilidad sin mitigación en un sistema de producción. Los agentes de amenazas pueden usar este tipo de vulnerabilidad para atacar el sistema. Los desarrolladores suelen darse cuenta de la vulnerabilidad a raíz del ataque.

### aviso de tiro cero

Proporcionar a un [LLM](#) instrucciones para realizar una tarea, pero sin ejemplos (imágenes) que puedan ayudar a guiarla. El LLM debe utilizar sus conocimientos previamente entrenados para

realizar la tarea. La eficacia de las indicaciones cero depende de la complejidad de la tarea y de la calidad de las indicaciones. [Consulte también las indicaciones de pocos pasos.](#)

#### aplicación zombi

Aplicación que utiliza un promedio de CPU y memoria menor al 5 por ciento. En un proyecto de migración, es habitual retirar estas aplicaciones.

Las traducciones son generadas a través de traducción automática. En caso de conflicto entre la traducción y la versión original de inglés, prevalecerá la versión en inglés.